

MODULHANDBUCH

der Master-Studiengänge im Fachbereich Informatik
Prüfungsordnung 2021

Inhaltsverzeichnis

Hinweise zu den Modulhandbucheinträgen	3
Externe Module.....	3
Abschlussarbeit	4
Advanced Game Technology.....	5
Anforderungsmanagement	6
Architektur von Cloud-Anwendungen.....	7
Berechenbarkeit und Komplexität	9
Data Science.....	10
Data Warehouse	11
Fortgeschrittene Methoden der Computergrafik.....	12
Ganzzahlige Lineare Optimierung	13
Geschäftsprozessmanagement	14
High Performance Computing	15
Implementierung von ERP-Systemen	16
Informationssicherheit	17
Interactive Physical Simulation	19
Kooperative Systeme	20
Künstliche Intelligenz für Roboter	21
Künstliche neuronale Netze	22
Lineare Optimierung	23
Maschinelles Lernen	24
Mensch-Computer-Interaktion.....	25
Mobile Robotik.....	26
Projektstudium	27
Safety KI-basierter Cyber Physical Systems.....	28
Seminar	29
Simulationstechnik und Reinforcement Learning	30
Software-Architekturen.....	31
Software-Qualitätsmanagement	32
Statistics and Learning Theory	33
Ubiquitous Computing	34
Verlässliche Echtzeitsysteme	35
Verteilte Systeme.....	36
Visuelle Navigation	37

Hinweise zu den Modulhandbucheinträgen

- Die Dauer aller Module beträgt jeweils ein Semester.
- Die zeitliche Lage aller Module ergibt sich aus Anlage 1 bzw. Anlage 2 der Fachprüfungsordnung.
- Neben den jeweils genannten Voraussetzungen werden in allen Modulen Grundkenntnisse der Informatik erwartet, die in der Regel in einem Bachelor-Studium der Informatik vermittelt werden.
- Zur individuellen Vorbereitung kann der Besuch der im Feld „Empfohlene Voraussetzungen“ genannten Module dienen, ebenso wie die Angaben im Feld „Literatur“. Weitere Literaturangaben finden sich ggf. in den Vorlesungsunterlagen, die i.d.R. auf Stud.IP (<https://studip.hochschule-trier.de>) zur Verfügung gestellt werden.
- Falls in einem Modul eine Studienleistung zu erbringen ist, ist diese gemäß § 6 der allgemeinen Prüfungsordnung Voraussetzung für die Zulassung zur Prüfung. Die semesteraktuelle Form der Studienleistung wird zu Beginn der Lehrveranstaltung bekannt gegeben.
- Bei Angabe mehrerer alternativer Prüfungsformen für ein Modul wird die semesteraktuelle Prüfungsform zu Beginn der Lehrveranstaltung bekannt geben.
- Voraussetzung für die Vergabe von ECTS-Punkten ist eine Bewertung der Prüfungsleistung mit mindestens der Note „ausreichend“.
- Die Gewichtung eines Modulergebnisses zur Bildung der Gesamtnote ergibt sich gemäß § 10 der Fachprüfungsordnung aus dem Verhältnis der ECTS-Punkte für das Modul und der Gesamtanzahl der ECTS-Punkte.

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 Technik

- Projektmanagement

Abschlussarbeit			
Inhalte	Die Abschlussarbeit umfasst die Bearbeitung einer qualifizierten Aufgabenstellung deren Schwierigkeitsgrad der späteren Berufspraxis eines Master-Absolventen entspricht mithilfe wissenschaftlicher Methoden. Die Aufgabenstellung kann theoretische, experimentelle, empirische oder praxisorientierte Probleme umfassen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken eingeübt sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens sowie in einem Kolloquium vor einer Prüfungskommission. Dabei wird der Inhalt der Abschlussarbeit im Kontext des jeweiligen Themas hinterfragt.		
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist ein größeres Fachproblem selbstständig mit wissenschaftlichen Methoden zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen, einschließlich der wissenschaftlichen Grundlagen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf neue Fragestellungen zu übertragen und darüber hinaus selbstständig um relevante Inhalte zu erweitern, zu bewerten und wissenschaftlich zu interpretieren. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Wissenschaft entsprechende Lösung für das Fachproblem. Sie können ihre Ergebnisse darüber hinaus professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren, in einem Kolloquium darlegen und argumentativ vertreten.		
Lehrform	<input type="checkbox"/> Vorlesung		
	<input type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input checked="" type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Seminar“ und „Projektstudium“. Weitere empfohlene Voraussetzungen abhängig von der Aufgabenstellung; werden vom Betreuer festgelegt.		
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	30	30 Stunden	870 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	13.02.2023		

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: <ul style="list-style-type: none"> ▪ 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	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Einführung in die Computergrafik“ und „C/C++-Programmierung“		
Literatur	<ul style="list-style-type: none"> ▪ T. Akenine-Möller, E. Haines, N.Hoffman: Real-Time Rendering. 4th Edition, AK Peters, 2018. ▪ A.Louis, P. Maaß, A. Rieder: Wavelets: Theorie und Anwendungen. Vieweg+Teubner Verlag, 1994. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> 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	13.02.2023		

Anforderungsmanagement		
Inhalte	<p>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.</p> <p>Im zweiten Teil der Vorlesung wird auf aktuelle Entwicklungen im Bereich des Anforderungsmanagement eingegangen. Hierbei wird der Zusammenhang zwischen Anforderungsmanagement und Variantenmanagement behandelt. Im Mittelpunkt steht zunächst das Verstehen der variantenbezogenen Entwicklungstechniken (Domänen- und Applikationsengineering). Das Wissen, Verstehen und die Anwendung werden durch praxisorientierte Übungen, sowie den Einsatz aktueller Anforderungsmanagement- und Variantenmanagement-Werkzeuge vertieft.</p>	
Lernergebnisse	<p>Die Studierenden haben aufbauend auf der gleichnamigen Bachelor-Vorlesung die Kenntnisse bzgl. der Modellierung, Ermittlung und Dokumentation von Anforderungen vertieft. Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Systeme hinsichtlich Ihrer wesentlichen Eigenschaften erfassen und abstrahieren, ▪ Anforderungen (funktional, nicht-funktional) an Systeme sammeln und spezifizieren, ▪ Techniken zur Stakeholder-Identifikation und Interviews kennen und anwenden, ▪ Review-Arten unterscheiden und anwenden, ▪ Anforderungen auf deren Qualität beurteilen, ▪ Agile Vorgehensweisen im Requirements Engineering unterscheiden und anwenden, ▪ aktuelle Entwicklungen wie Standardisierungen im Anforderungsmanagement (z.B. ReqIF) beurteilen, und ▪ aktuelle Werkzeuge aus dem Bereich Variantenmanagement im Requirements Engineering anwenden. 	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input checked="" type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Grundlagen des Anforderungsmanagements“	
Literatur	<ul style="list-style-type: none"> ▪ Ian Sommerville: Software Engineering. Pearson Studium, 2007. ▪ Klaus Pohl: Requirements Engineering - Grundlagen, Prinzipien, Techniken. dpunkt.Verlag, 2. Auflage, 2008. ▪ Soren Lauesen: Software Requirements, Styles and Techniques. Addison-Wesley, 2002. ▪ Colin Hood, Simon Wiedemann, Stefan Fichtinger, Urte Pautz: Requirements Management. Springer, 2008. ▪ Helmut Partsch: Requirements-Engineering systematisch. Springer, 2. Auflage, 2010. 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. G. Rock	
Modulverantwortliche(r)	Prof. Dr. G. Rock	
Änderungsdatum	03.02.2023	

Architektur von Cloud-Anwendungen	
Inhalte	<p>Wer bisher Cloud Computing mit einem Dienst wie Dropbox gleichgesetzt hat, bei dem es lediglich um das Speichern von Daten „in der Cloud“ geht, wird in diesem Modul eines Besseren belehrt. In diesem Modul wird deutlich, dass Cloud Computing die Möglichkeit bietet, komplette Anwendungen, die traditionell in den Rechenzentren von Firmen (und Hochschulen) betrieben werden, „in die Cloud“ zu verlagern. Aber auch neu gegründeten Firmen bietet Cloud Computing eine gute Basis zum Starten ihrer Geschäfte ohne größere Investitionen zum Aufbau einer Rechnerinfrastruktur. Viele Firmen wie z.B. Netflix nutzen diese Möglichkeit heute schon sehr intensiv.</p> <p>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.</p> <p>U.a. werden folgende Themen behandelt, wobei bei allen Themen die AWS-Dienste als Beispiele herangezogen werden:</p> <ul style="list-style-type: none"> ▪ 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	<p>Die Studierenden besitzen nach dem Absolvieren dieses Moduls folgende Fähigkeiten:</p> <ul style="list-style-type: none"> ▪ Die Studierenden können die grundlegenden Konzepte des Cloud Computing sowie deren Vor- und Nachteile erläutern. ▪ Die Studierenden können einen Überblick über die unterschiedlichen Cloud-Dienste anhand der AWS-Dienste von Amazon geben. Sie können die Besonderheiten dieser Dienste, aber auch ihre Begrenzungen darstellen. ▪ Die Studierenden sind in der Lage, komplette, problemspezifische Anwendungsarchitekturen zu entwerfen und diese mit Hilfe der AWS-Dienste aufzubauen. Dabei werden Entwurfsmuster und Best Practices verwendet, so dass die Aspekte Sicherheit, Hochverfügbarkeit, Lastverteilung und Fehlertoleranz besondere Berücksichtigung finden. ▪ Die Studierenden können ihre eigenen Lösungen einem Publikum gegenüber präsentieren sowie die vorgestellten Lösungen ihrer Mitstudierenden kritisch bewerten.
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt
Empfohlene Voraussetzungen	Keine
Literatur	Originalunterlagen von Amazon AWS (werden Studierenden zur Verfügung gestellt)
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)
Verwendbarkeit	Künstliche Intelligenz und Data Science <input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF

Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. R. Oechsle		
Modulverantwortliche(r)	Prof. Dr. R. Oechsle		
Änderungsdatum	22.11.2022		

Berechenbarkeit und Komplexität			
Inhalte	<ul style="list-style-type: none"> ▪ Äquivalente Modelle der Computer-Berechenbarkeit: WHILE, Turingmaschinen ▪ Algorithmenbegriff, These von Church ▪ Berechenbare Funktionen und ihre grundlegenden Eigenschaften, z.B. s-m-n-Theorem, universelle Algorithmen, Gödelisierung ▪ Entscheidbarkeit, Semi-Entscheidbarkeit Aufzählbarkeit, äquivalente Charakterisierungen, die Klassen REC und RE ▪ Unentscheidbare Mengen, Halteproblem, Many-one-Reduzierbarkeit, RE-Vollständigkeit, Beispiel nicht-aufzählbarer Mengen, Reduktionsmethode, Satz von Rice ▪ Laufzeitanalyse, Polynomial- und Exponentialzeit, Katalog klassischer Entscheidungsprobleme ▪ Die Klassen P und NP, NP-Vollständigkeit, originärer Vollständigkeitsbeweis ▪ Reduktionen durch Komponentendesign, Vollständigkeitsbeweise für Auswahl-, Graphen- und Reihenfolgeprobleme ▪ Strenge NP-Vollständigkeit, Pseudo-Polynomialzeit, Large Number Problems, Dynamische Programmierung, pseudo-polynomielle Reduktionen ▪ NP-Optimierungsprobleme, die Klassen PO und NPO, Turing-Reduzierbarkeit, Begriff der (strengen) NP-Härte, PO-NPO-Frage ▪ Approximationsalgorithmen, Performanzrate, Beispiele mit konstanter Performanzrate, Approximierbarkeit, Approximationsschema 		
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Berechenbarkeitsmodelle hinsichtlich ihrer prinzipiellen Leistungsfähigkeit einordnen, ▪ die Bedeutung eines präzisen Algorithmenbegriffs erläutern, ▪ Berechnungsprobleme hinsichtlich ihrer prinzipiellen algorithmischen Beherrschbarkeit analysieren und bewerten, ▪ die Bedeutung der P-NP-Theorie präsentieren und Berechnungsprobleme in dieser Hinsicht klassifizieren, ▪ die Methode der Reduktion auf neue Berechnungsprobleme anwenden und ▪ Optimierungsprobleme kategorisieren und deren praktische Lösbarkeit beurteilen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ H. Rogers Jr.: Theory of Recursive Functions and Effective Computability. MIT-Press, 1967. 3rd printing 1992. ▪ Michael R. Garey and David S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979. ▪ K. W. Wagner: Theoretische Informatik – Eine kompakte Einführung. Springer, 2nd edition, 2003. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	22.11.2022		

Data Science		
Inhalte	<ul style="list-style-type: none"> ▪ Grundlagen und Anwendungsfälle für Data Science ▪ Daten <ul style="list-style-type: none"> ▪ typische Datenquellen im Unternehmen ▪ Arten von Daten: Stammdaten, Bewegungsdaten ▪ strukturierte, unstrukturierte und semistrukturierte Daten ▪ Netzwerkdaten/Graphen ▪ Werkzeuge zur Datenanalyse und -vorverarbeitung ▪ Datenqualität ▪ Datenverarbeitung <ul style="list-style-type: none"> ▪ Architektur von Datenverarbeitungsstrecken ▪ Umgang mit großen Datenmengen ▪ Datenintegration ▪ Data Ingestion ▪ Präsentationsschicht ▪ Business Intelligence <ul style="list-style-type: none"> ▪ Grundlagen Data Warehousing ▪ Reporting ▪ Visualisierung ▪ Empfehlungssysteme ▪ Social Network Analysis 	
Lernergebnisse	<p>Die Studierenden sind nach der Teilnahme in der Lage</p> <ul style="list-style-type: none"> ▪ vorhandene Datenbestände in Unternehmen zu bewerten, zu verarbeiten und zu nutzen, ▪ die Qualität und Struktur von Daten einzuschätzen, ▪ Daten aus verschiedenen Quellen aufzubereiten und zu integrieren, ▪ Daten und daraus abgeleitetes Wissen für Fachanwender zu präsentieren, ▪ gängige Werkzeuge wie Kafka, Camel, PDI, Pandas, Jupyter einzusetzen und ▪ Wissen und Empfehlungen aus sozialen Netzwerken zu generieren. <p>Die Vorlesung ist im Zusammenspiel mit der Veranstaltung „Maschinelles Lernen“ konzipiert, in der konkrete Lernverfahren vermittelt werden.</p>	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Datenbanken“ und „Big-Data-Technologien“	
Literatur	<ul style="list-style-type: none"> ▪ Cady, F.: The Data Science Handbook. Wiley, 2017. ▪ O’Neil, C., Schutt, R.: Doing Data Science. O’Reilly, 2014. ▪ Kotu, V., Deshpande, B.: Data Science: Concepts and Practice. 2. Auflage, Morgan Kaufmann, 2019. ▪ Barabasi, A.-L., Posfai, M.: Network Science. Cambridge University Press, 2016. ▪ Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, 2016. ▪ Kimball, R., Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013. 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz	
Modulverantwortliche(r)	Prof. Dr. C. Schmitz	
Änderungsdatum	22.11.2022	

Data Warehouse			
Inhalte	<ul style="list-style-type: none"> ▪ Motivation zum Einsatz von Data Warehouses (DWH) ▪ Einbettung von DWH in typische IT-Landschaften und -Prozesse ▪ SQL und PL/SQL ▪ Dimensionale Modellierung von DWH nach Kimball ▪ Implementierung von DWH in relationalen Datenbanksystemen ▪ OLAP und analytische Operatoren in SQL ▪ ETL-Prozesse (Extract-Transform-Load) und ETL-Tools ▪ Semistrukturierte Daten in XML und JSON 		
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ die Motivation für den Einsatz von DWH und typische Einsatzszenarien für DWH in IT-Landschaften erklären, ▪ geeignete Granularität, Dimensionen, Fakten und Metriken für gegebene Anwendungsszenarien von DWH auswählen, ▪ dimensionale Modelle nach Kimball in einer relationalen Datenbank implementieren, ▪ fortgeschrittene analytische Anfragen gegen relationale Datenbanken in SQL stellen, ▪ Daten mithilfe von PL/SQL, AWK und graphischen Werkzeugen wie z. B. PDI extrahieren und transformieren, sowie ▪ die Unterschiede zwischen semistrukturierten und strukturierten Daten verstehen und deren Verbindung innerhalb einer relationalen Datenbank umsetzen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Datenbanken“		
Literatur	<ul style="list-style-type: none"> ▪ William Inmon: Building the Data Warehouse. 4. Auflage, Wiley, 2005. ▪ Ralph Kimball, Margy Ross: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz		
Modulverantwortliche(r)	Prof. Dr. C. Schmitz		
Änderungsdatum	04.01.2023		

Fortgeschrittene Methoden der Computergrafik			
Inhalte	<ul style="list-style-type: none"> ▪ 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	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	A. P. Godse, Dr. D. A. Godse: Computer Graphics – Concepts and Algorithms. 2020, Technical Publications.		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> 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	01.02.2023		

Ganzzahlige Lineare Optimierung			
Inhalte	<ul style="list-style-type: none"> ▪ Ganzzahlige Lineare Programme [MIP, IP, BIP] ▪ Beispielprobleme und deren Modellierung, Modellierungstechniken, Komplexitätsbetrachtung ▪ Formulierungen und optimale LP-Relaxierungen, vollständig unimodulare Matrizen ▪ LP-basiertes Branch-and-Bound-Verfahren ▪ Spaltengenerierung, Branch-and-Price ▪ Schnittebenenverfahren, Branch-and-Cut ▪ Praktischer Einsatz von LP-Solvern ▪ Laborprojekt 		
Lernergebnisse	Die Studierenden können <ul style="list-style-type: none"> ▪ kombinatorische Optimierungsprobleme als MIPs modellieren, ▪ die zugrundeliegende Theorie der ganzzahligen Linearen Programme erläutern, ▪ die algorithmischen Verfahren darstellen und gegenüberstellen, ▪ die behandelten Lösungsmethoden für Beispielprobleme auswählen und anwenden sowie ▪ LP-Solver praktisch einsetzen und mit deren Hilfe schwierige Optimierungsprobleme lösen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input checked="" type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Kenntnisse der Linearen Optimierung (LP-Modellierung, Simplex-Verfahren, Dualität)		
Literatur	<ul style="list-style-type: none"> ▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997. ▪ Jiri Matousek and Bernd Gärtner: Understanding and Using Linear Programming. Universitext. Springer Science & Business Media, July 2007. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	22.11.2022		

Geschäftsprozessmanagement			
Inhalte	<ul style="list-style-type: none"> ▪ Definitionen und Begriffsklärungen im Kontext Geschäftsprozessmanagement (GPM) ▪ Strategisches und operatives GPM ▪ Phasen des sog. GPM-Zyklus ▪ GPM-Modellierungsmethoden: <ul style="list-style-type: none"> ▪ 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	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ den Sinn und Einsatzpotentiale von GPM-Systemen als Mittel zur erfolgreichen Umsetzung eines ganzheitlichen Prozessmanagements verstehen, ▪ Geschäftsprozesse mit Standard-Notationen beschreiben, ▪ wesentliche Prozessablaufmuster (Workflow Patterns) beschreiben und modellieren, ▪ Funktionalitäten existierender GPM-Systeme kennen und kritisch beurteilen, ▪ den GPM-Zyklus mit Hilfe eines GPM-Werkzeugs in den Übungen exemplarisch durchlaufen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers: Fundamentals of Business Process Management. Springer, 2018. ▪ J. Freund, B. Rücker: Praxishandbuch BPMN - Mit Einführung in DMN, 6. Auflage, Hanser, 2019. ▪ Wil M.P. van der Aalst: Process Mining: Data Science in Action. 2. Auflage, Springer, 2016. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. A. Lux		
Modulverantwortliche(r)	Prof. Dr. A. Lux		
Änderungsdatum	18.03.2024		

High Performance Computing			
Inhalte	<ul style="list-style-type: none"> ▪ General paradigms for parallel programming ▪ Using the Threading Building Blocks Library for CPU parallelism <ul style="list-style-type: none"> ▪ Concepts of the task stealing scheduler ▪ Efficient memory management for parallel systems ▪ General parallelism concepts in TBB ▪ Using CUDA for GPGPU computing <ul style="list-style-type: none"> ▪ 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	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> 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).		
Literatur	<ul style="list-style-type: none"> ▪ Michael McCool, Arch D. Robinson, James Reinders: Structured Parallel Programming. Morgan Kaufmann, 2012. ▪ James Reinders: Intel Threading Building Blocks. O'Reilly, 2007. ▪ Jason Sanders, Edward Kandrot: CUDA by Example. Addison-Wesley Professional, 2010. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	22.11.2022		

Implementierung von ERP-Systemen			
Inhalte	<ul style="list-style-type: none"> ▪ 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	<input type="checkbox"/> Vorlesung		
	<input type="checkbox"/> Übung		
	<input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Objektorientierte Programmierung - Einführung“, „Datenbanken“ und „Produktionswirtschaft“		
Literatur	<ul style="list-style-type: none"> ▪ Thorsten Franz, Karl-Heinz Kühnhauser: Einstieg in ABAP. 6. Auflage, 2023, Rheinwerk. ▪ Paul Fuchs: SQL: Handbuch für Einsteiger. 2020, BMU Verlag. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> 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	07.03.2024		

Informationssicherheit		
Inhalte	<ul style="list-style-type: none"> ▪ Einführung: grundlegende Begriffe und Zusammenhänge ▪ Physische Sicherheit ▪ Zugriffskontrolle <ul style="list-style-type: none"> ▪ Authentisierungsverfahren und –techniken ▪ Zugriffskontrollmodelle ▪ Rechtevergabe ▪ Organisatorische, konzeptionelle und prozessorientierte Sicherheit <ul style="list-style-type: none"> ▪ 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	<p>Die Studierenden haben einen gesamtheitlichen Einblick in die Informationssicherheit erlernt und verstanden. Dies umfasst neben technischen insbesondere auch physische, organisatorische, konzeptionelle und rechtliche Aspekte. Die Studierenden können</p> <ul style="list-style-type: none"> ▪ strukturierte Vorgehensweisen zur Umsetzung von Informationssicherheit anhand von Standards wie z.B. ISO 27x oder BSI 100 benennen und anwenden, ▪ Risikoanalysen durchführen, ▪ die Historie von ausgewählter Malware im Detail aufzeigen sowie Algorithmen zur Malware-Erkennung beschreiben und in Beispielaufgaben anwenden, ▪ fortgeschrittene Verfahren zur Authentisierung und Autorisierung beschreiben (wie z.B. die biometrische Authentisierung über den Fingerabdruck oder RBAC) und in Beispielaufgaben anwenden und berechnen, sowie ▪ die Grundprinzipien und Standards des Patch Managements beschreiben und in Beispielaufgaben anwenden. 	
Lehrform	<input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input checked="" type="checkbox"/> Labor (Ausgewählter Vorlesungsstoff und die Übungen werden vertieft durch praktische Übungen.) <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Besuch einer einführenden Veranstaltung zur IT-Sicherheit im Bachelor-Studium	
Literatur	<ul style="list-style-type: none"> ▪ Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons ▪ John Aycock, Computer Viruses and Malware, Springer ▪ Bundesamt für Sicherheit in der Informationstechnik, Standardreihe 200-x ▪ Claudia Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle, Oldenbourg ▪ Shon Harris, CISSP Certification All-in-one Exam Guide: Complete coverage of all Certified Information Systems Security Professional domains, McGraw-Hill ▪ ISO 27001 und ISO 27002 ▪ Norbert Pohlmann: Cyber-Sicherheit, Springer, 2019 [Als eBook der Bibliothek vorhanden] ▪ Sowa: Management der Informationssicherheit, Springer 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	75 Stunden
		Selbststudium
		105 Stunden
Lehrende[r]	Prof. Dr. K. Knorr	

Modulverantwortliche(r)	Prof. Dr. K. Knorr
Änderungsdatum	22.11.2022

Interactive Physical Simulation			
Inhalte	<ul style="list-style-type: none"> ▪ Collision detection <ul style="list-style-type: none"> ▪ 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 <ul style="list-style-type: none"> ▪ Collision reaction between two rigid bodies with one point of contact ▪ Multiple rigid bodies and multiple points of contact ▪ Impulse base Simulation methods ▪ Constraint based Simulation methods <ul style="list-style-type: none"> ▪ Early methods like penalty and verlet integration ▪ Lagrange Dynamics for constrained motion ▪ linear complimentary problems ▪ Simulation of breaking and tearing for games with lagrange dynamics ▪ Coupling of Simulation and Animation for articulated characters 		
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	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Competences according to the learning outcome of the bachelor modules „Lineare Algebra“ (Linear Algebra) and „Angewandte Mathematik“ (Applied Mathematics).		
Literatur	<ul style="list-style-type: none"> ▪ H. R. Schwarz: Numerische Mathematik. Springer Vieweg, 2011. ▪ David H. Eberly: Game Physics. Taylor & Francis, 2. Auflage, 2010. ▪ Gino van den Bergen: Collision Detection in Interactive 3D Environments. CRC Press, 2003. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	22.11.2022		

Kooperative Systeme			
Inhalte	<ul style="list-style-type: none"> ▪ Soziale und technische Aspekte der Kooperation ▪ Klassifikation von Groupware-Lösungen ▪ Gestaltung von Groupware-Lösungen ▪ Awareness, Kommunikations- und Koordinationsunterstützung ▪ Cognitive Systems Engineering ▪ Mensch-Maschine-Kooperation ▪ Geteilte Kontrolle und Entscheidungsunterstützung ▪ Symbolische und subsymbolische Ansätze künstlicher Intelligenz ▪ Intelligent User Interfaces ▪ Mensch-Maschine-Integration 		
Lernergebnisse	Die Studierenden können <ul style="list-style-type: none"> ▪ Grundlagen, Prinzipien und Anwendungsmöglichkeiten computergestützter kooperativer Arbeit erläutern, ▪ die wichtigsten Theorien und Modelle der Mensch-Maschine-Kooperation beschreiben, und ▪ kooperative Systeme analysieren, konzipieren, realisieren und evaluieren. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ T. Gross & M. Koch: Computer-Supported Cooperative Work. Oldenbourg, 2007. ▪ D. Coleman: Groupware. Prentice-Hall, 1997. ▪ G. Schwabe et al. (Hrsg.): CSCW-Kompodium. Springer, 2001. ▪ M. Beaudouin-Lafon (Hrsg.): Computer-Supported Cooperative Work. Wiley, 1998. ▪ C. Reuter: Sicherheitskritische Mensch-Computer-Interaktion. Interaktive Technologien und Soziale Medien im Krisen- und Sicherheitsmanagement. 2018 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	22.11.2022		

Künstliche Intelligenz für Roboter		
Inhalte	<ul style="list-style-type: none"> ▪ Entscheidungsfindung und Planung für intelligente Roboter und Kognitive Systeme ▪ Probabilistisches Schlussfolgern ▪ Inferenz ▪ Parameterschätzung ▪ Bayessche Netze und Lernen von Strukturen ▪ Einfache und komplexe Entscheidungen ▪ Sequenzielle Probleme ▪ Entscheidungsfindung unter Unsicherheit ▪ Modellbasierte und Modellfreie Methoden ▪ Verhalten bei Zustandsunsicherheit ▪ Multi-Agenten-Systeme und Kollaboration 	
Lernergebnisse	Die Studierenden können <ul style="list-style-type: none"> ▪ stochastische Modelle erstellen und implementieren, ▪ verstehen, was Bayessche Netze sind und diese auf verschiedene Probleme anwenden, ▪ Parameterschätzung nach Bayes anwenden und Explorationsstrategien herleiten, ▪ MAP-Schätzer verstehen und implementieren, ▪ MDPs und POMDPs anwenden, ▪ Q-Learning verstehen und anwenden, ▪ Dynamische Programme formulieren, ▪ Suchstrategien einordnen und effizient einsetzen, ▪ lokale Optimierungsstrategien anwenden und um Heuristiken ergänzen, sowie ▪ ggf. Planungsmethoden theoretisch verstehen und praktisch anwenden. 	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Grundlagenkenntnisse in Stochastik und Maschinellem Lernen sind von Vorteil, werden aber nicht vorausgesetzt.	
Literatur	<ul style="list-style-type: none"> ▪ S. M. LaValle: Planning Algorithms. Cambridge University Press, 2006. ▪ M. J. Kochenderfer et al.: Algorithms for Decision Making. MIT Press, 2022. ▪ R. Neapolitan: Learning Bayesian Networks. Prentice Hall, 2003. ▪ S. Russell and P. Norvig: Artificial Intelligence: A Modern Approach. Pearson, 4. Auflage (2020). 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. J. Graf	
Modulverantwortliche(r)	Prof. Dr. J. Graf	
Änderungsdatum	25.01.2023	

Künstliche neuronale Netze			
Inhalte	<p>Künstliche neuronale Netze (KNN) sind eine zentrale Klasse von Funktionsmodellen, die in sämtlichen Anwendungsgebieten des maschinellen Lernens eingesetzt werden. In diesem Modul werden, ausgehend von den Anfängen der KNN, der Aufbau, das Trainieren und zentrale Architekturen eingeführt. Dabei werden auch wesentliche Punkte der konkreten Auswertung und Analyse besprochen. Weiterhin werden einige Aspekte der Theorie zu KNN insbesondere mit Bezug zu dem Gebiet der <i>Learning Theory</i> besprochen.</p> <p>Folgende Inhalte werden in diesem Modul behandelt:</p> <ul style="list-style-type: none"> ▪ Architekturen: MLP, CNN, RNN, LSTM, Autoencoder, Attention, Transformer ▪ Algorithmen: Perzeptron-Algorithmus, Backpropagation, SGD und Varianten ▪ Theorie KNN: Approximationseigenschaften, VC-Dimension und Generalisierung von KNN 		
Lernergebnisse	<p>Studierende erlernen tieferegehende Konzepte, Methoden und Techniken im Bereich künstlicher neuronaler Netze, zur Durchführung anspruchsvoller Praxisprojekte in Anwendungsgebieten des maschinellen Lernens und Data Science. Studierende können sich selbstständig anhand von Primärliteratur, insbesondere in englischer Sprache, weitergehende wissenschaftliche Grundlagen erarbeiten und aneignen. Dabei erlernen die Studierenden auch sicherheitsrelevante und ethische Problemstellungen im Umgang mit KI-Methoden zu erkennen und kritisch einzuschätzen. Studierende können komplexe Problemstellungen analysieren und strukturieren, eigene Ideen zur Lösung entwickeln und die Lösung durch die Auswahl und Anwendung geeigneter Methoden erarbeiten.</p>		
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ I. Goodfellow et al.: Deep Learning. MIT Press, 2016. ▪ G. Strang: Linear Algebra and Learning from Data. Wellesley-Cambridge Press, 2019. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H.-P. Beise		
Modulverantwortliche(r)	Prof. Dr. H.-P. Beise		
Änderungsdatum	23.01.2023		

Lineare Optimierung			
Inhalte	<ul style="list-style-type: none"> ▪ Lineare Programme ▪ Beispielprobleme und deren Modellierung, Standardform, Slackform ▪ Simplexverfahren in Grundform, Nachweis der Korrektheit, Eindeutigkeit optimaler Lösungen, Initialisierung, Hauptsatz der Linearen Programmierung, ökonomische Interpretation der optimalen Slackform ▪ LP-Dualität, schwache und starke Dualität, Satz vom komplementären Schlupf, Interpretation der Dualisierung in ökonomischen Modellen ▪ Revidiertes Simplexverfahren, temporäre Schattenpreise, Duales Simplexverfahren ▪ Geometrie Linearer Programme, konvexe Polyeder ▪ Einsatz von LP-Solvern 		
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Berechnungsprobleme als lineare Programme modellieren, insbesondere Strukturvariablen identifizieren und Bedingungen mathematisch formulieren, ▪ die behandelten Verfahren reproduzieren und deren Korrektheit und Laufzeit erklären, ▪ Lineare Programme mit den behandelten Verfahren lösen und die erzielten Lösungen interpretieren, das Konzept der LP-Dualität wiedergeben und die zugrundeliegende Theorie begründen sowie den Einsatz von LP-Solvern am Beispiel praktisch umsetzen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Grundkenntnisse der linearen Algebra		
Literatur	<ul style="list-style-type: none"> ▪ Vasek Chvatal: Linear programming. A Series of Books in the Mathematical Sciences, 1983. ▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	22.11.2022		

Maschinelles Lernen		
Inhalte	<p>Die Methoden des maschinellen Lernens stellen die Grundlage für die aktuelle Forschung und Entwicklungen im Bereich intelligenter Systeme dar und sind integraler Bestandteile einer großen Anzahl gegenwärtiger, industrieller Anwendungen. Hierbei werden Modelle anhand vorliegender Daten trainiert, um spezifische Aufgabenstellung zu lösen.</p> <ul style="list-style-type: none"> ▪ Einführung und grundlegende Konzepte ▪ Datenvorverarbeitung / Merkmalsextraktion (Zeit- und Frequenzbereich) / Dimensionsreduktion ▪ Hauptachsentransformation (PCA) / Whitening ▪ Sammon-Transformation ▪ Multidimensional Skalierung (MDS), T-SNE ▪ Entscheidungsbäume ▪ Bayes Klassifikator ▪ k-means, k-Nearest Neighbour, Kerndichteschätzer ▪ Linear und nicht linear separierbare Probleme ▪ Quadratische Fehlerminimierung und Gradientenabstiegsverfahren ▪ Fischer lineare Diskriminanzanalyse (LDA) ▪ Linearer Klassifikator und generalisierte lineare Diskriminanzfunktionen ▪ Lineare und multiple lineare Regression ▪ Support Vector Machines ▪ Dynamic Time Warping ▪ Hidden-Markov-Modell ▪ Evaluierung von Lernverfahren ▪ Grundlagen neuronaler Netze 	
Lernergebnisse	Die Teilnehmer der Lehrveranstaltung kennen den aktuellen Forschungsstand im Bereich Maschinelles Lernen. Sie sind fähig, aktuelle Entwicklungen zu verstehen, zu analysieren und kritisch zu reflektieren.	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Keine	
Literatur	<ul style="list-style-type: none"> ▪ Duda, Richard O.: Pattern Classification: Pattern Classification. Wiley-Interscience. ▪ Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press. ▪ Ferdinand van der Heijden, de Ridder, Dick, Tax, David M. J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. Wiley. ▪ Bishop, Christopher M.: Neural Networks for Pattern Recognition. Oxford University Press. 	
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input checked="" type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. J. Lohscheller	
Modulverantwortliche(r)	Prof. Dr. J. Lohscheller	
Änderungsdatum	12.05.2023	

Mensch-Computer-Interaktion			
Inhalte	<ul style="list-style-type: none"> ▪ Grundlegende Begriffe der Mensch-Computer-Interaktion: Usability, User Experience, Accessibility ▪ Menschliche Informationsverarbeitung und Handlungsprozesse ▪ Ein-/Ausgabegeräte und Interaktionstechnologien ▪ Menschzentrierte Entwicklungsprozesse: ISO 9241-210, Contextual Design, Scenario-Based Design ▪ Usability und Software Engineering ▪ Methoden zur Analyse von Nutzungskontexten: Hierarchische Aufgabenanalyse, Personas, Prozessanalyse ▪ Low- und High-Fidelity-Prototyping ▪ Formative und summative Evaluationen ▪ Modelle für Mensch-Computer-Systeme ▪ Interaktion und Kooperation ▪ Normen und rechtliche Grundlagen 		
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Prinzipien und Methoden menschenzentrierter Entwicklung interaktiver Systeme erläutern. ▪ Kenntnisse über die menschliche Informationsverarbeitung in die Entwicklung interaktiver Systeme einbringen. ▪ Modelle und Kriterien zur Analyse und Bewertung interaktiver Systeme anwenden. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ Benyon, D. (2014). Designing interactive systems: A comprehensive guide to HCI, UX and interaction design (Third edition). Harlow, England: Pearson. ▪ Dahm, M. (2006). Grundlagen der Mensch-Computer-Interaktion. Informatik: Software-Ergonomie. München: Pearson Studium. ▪ Herczeg, M. (2018). Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme: De Gruyter. ▪ Reuter, C. (Hrsg.) (2018). Sicherheitskritische Mensch-Computer-Interaktion. Wiesbaden: Springer Fachmedien Wiesbaden. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	<input checked="" type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	22.11.2022		

Mobile Robotik		
Inhalte	<ul style="list-style-type: none"> ▪ Koordinatentransformationen für die Lokalisierung ▪ Inertialnavigation ▪ Kalibrierung von Beschleunigungssensoren und Drehratensensoren ▪ Technologie und Kalibrierung ▪ Algorithmen für Strap Down-Verfahren ▪ Satellitennavigation (Signale und Systeme) ▪ Erweiterungen (bspw. RTK) ▪ Datenfusion von INS/GNSS (loosely- und tightly-coupled-Architekturen) in Theorie (Error State) und Praxis (Simulationsumgebung) ▪ Visuelle Navigation am Beispiel von Visueller Odometrie (VO) ▪ Möglichkeiten zur Integration von Inertialen VO und INS/GNSS/VO 	
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ geodätische Transformationen korrekt anwenden, ▪ Fehlermodelle für die Kalibrierung von Inertialen Systemen formulieren, ▪ Strap Down-Verfahren implementieren, ▪ Satellitennavigation auf Komponentenebene erklären, ▪ Lokalisierung mit Satelliten in Theorie und Praxis anwenden, ▪ Datenfusionsarchitekturen implementieren (loosely und tightly-coupled), ▪ die Grundlagen von Visueller Odometrie verstehen und implementieren, und ▪ INS/GNSS/VO-Architekturen entwerfen und diskutieren. 	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Einführung in die Robotik“, „Robotersehen“ und „Kognitive Sichtsysteme“	
Literatur	<ul style="list-style-type: none"> ▪ R. Siegwart et al: Autonomous Mobile Robotics. Springer, 2. Auflage (2011). ▪ J. Hertzberg et al: Mobile Robotik. Springer, 2012. ▪ Groves: Navigation Systems. Artech House, 2013. ▪ Wendel: Integrierte Navigationssysteme. Oldenbourg, 2011. ▪ Jekeli: Inertial Navigation Systems with Geodetic Applications. De Gruyter, 2000. ▪ R. M. Rogers: Applied Mathematics in INS. Transatlantic Publishers, 2007. ▪ Titterton and Weston: Strapdown Inertial Navigation Technology. IET, 2. Auflage (2005). ▪ J. Graf: Vorlesungsunterlagen zum Modul Visuelle Navigation 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. J. Graf	
Modulverantwortliche(r)	Prof. Dr. J. Graf	
Änderungsdatum	25.01.2023	

Projektstudium		
Inhalte	Das Projektstudium umfasst die Bearbeitung einer qualifizierten Aufgabenstellung aus der Praxis unter intensiver Betreuung durch einen Dozenten. Der Schwierigkeitsgrad entspricht der späteren Berufspraxis eines Master-Absolventen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens.	
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist sich selbständig in ein Anwendungsgebiet der Informatik einzuarbeiten und ein praktisches Problem zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf vorgegebene Fragestellungen anzuwenden. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Technik entsprechende Lösung für das praktische Problem. Sie verstehen die Erfolgsfaktoren und beherrschen Methoden der Definition, Planung sowie Kontrolle von Projekten. Daher können sie komplexe Aufgaben sinnvoll strukturieren und typische Schnittstellenprobleme sowohl auf technisch-fachlicher als auch auf sozialer Ebene bewältigen. Sie können Arbeitsergebnisse professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren.	
Lehrform	<input type="checkbox"/> Vorlesung <input type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input checked="" type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; wird vom Betreuer festgelegt	
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.	
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	15	30 Stunden
		Selbststudium
		420 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik	
Modulverantwortliche(r)	Fachrichtungsleiter Informatik	
Änderungsdatum	13.02.2023	

Safety KI-basierter Cyber Physical Systems		
Inhalte	<ul style="list-style-type: none"> ▪ Definition Cyber Physical Systems (CPS) ▪ Terminologie zu Safety, Dependability, Fault-Tolerance, Real-Time ▪ Beispiele KI-basierter Cyber Physical Systems, insbesondere aus den Bereichen Advanced Driver Assistance Systems (ADAS) und Automated Driving Systems (ADS) ▪ Zusammenhang Safety und Security ▪ Herausforderung Safety und KI ▪ Gesellschaftliche Bedeutung und ethische Fragestellungen ▪ Safety of the Intended Function (SOTIF) vs Functional Safety (FuSa) ▪ Einschlägige Standards und Normen (z.B. SAE J 3016, ISO 26262, ISO/DIS 21448) ▪ Übungsprojekt am Forschungsgrößgerät dynamischer Fahrsimulator FaSiMo: KI für ADAS bzw. ADS entwickeln gemäß SOTIF 	
Lernergebnisse	<p>Die Studierenden können die besonderen Herausforderungen bei der Entwicklung sicherheitsrelevanter KI-basierter Cyber Physical Systems erläutern. Sie können erklären, welche Rolle unterschiedliche Sicherheitsnormen bei der Entwicklung KI-basierter Lösungen für das automatisierte Fahren spielen und was der Unterschied zwischen SOTIF und FuSa ist. Sie kennen die grundlegenden Schritte, die bei der Entwicklung von Deep Learning/KI-Lösungen erforderlich sind, wissen um die Limitierungen solcher Systeme hinsichtlich Safety und kennen die standardisierten Herangehensweisen gemäß SOTIF. Sie sind in der Lage in interdisziplinären Teams unter Verwendung der einschlägigen Fachbegriffe Lösungsansätze für sicherheitsrelevante KI-basierte CPS gemäß dem SOTIF Prozess zu diskutieren, analysieren, entwerfen und zu gestalten.</p>	
Lehrform	<input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Gute Kenntnisse der englischen Sprache; Programmierkenntnisse (z.B. in Python oder C/C++)	
Literatur	<ul style="list-style-type: none"> ▪ ISO 21448 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avizienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013. 	
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider	
Modulverantwortliche(r)	Prof. Dr. J. Schneider	
Änderungsdatum	06.02.2023	

Seminar			
Inhalte	Das Seminar umfasst die selbstständige Erarbeitung eines vorgegebenen begrenzten Themenbereiches anhand von wissenschaftlicher Fachliteratur und anderen Quellen sowie dessen schriftliche und mündliche Darstellung. Es werden wechselnde aktuelle Themen aus der Informatik angeboten, die im Schwierigkeitsgrad für das Master-Studium angemessen sind.		
Lernergebnisse	Die Studierenden verfügen über ein vertieftes Verständnis spezifischer Theorien sowie Ansätze im gewählten Themenbereich und kennen den jeweils aktuellen Stand der Technik. Sie haben eigenständige Lernstrategien erfolgreich angewandt sowie ihre Kommunikations- und Kooperationsfähigkeit in Teams verbessert. Sie kennen wesentliche Merkmale von Präsentationstechniken und haben diese bei der verständlichen Darstellung ihrer Ergebnisse angewendet.		
Lehrform	<input type="checkbox"/> Vorlesung		
	<input type="checkbox"/> Übung		
	<input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Empfohlene Voraussetzungen abhängig vom Thema des Seminars; werden vom Betreuer festgelegt.		
Literatur	Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH.		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input checked="" type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	3	18 Stunden	72 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	13.02.2023		

Simulationstechnik und Reinforcement Learning		
Inhalte	<ul style="list-style-type: none"> ▪ Modelling und Simulation <ul style="list-style-type: none"> ▪ Projektablauf ▪ Diskrete Simulation ▪ Kontinuierliche Simulation ▪ Empirische Modelle ▪ Reinforcement Learning <ul style="list-style-type: none"> ▪ Grundlagen ▪ Tabellenverfahren ▪ Approximation mit Neuronalen Netzen ▪ Actor Critic ▪ Alpha Zero 	
Lernergebnisse	Die Studierenden können <ul style="list-style-type: none"> ▪ die Grundkonzepte von Modelling und Simulation sowie Reinforcement Learning erklären, ▪ die vorgestellten Techniken programmieren, sowie ▪ die vorgestellten Techniken in einem praktischen Projekt umsetzen. 	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input checked="" type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	<ul style="list-style-type: none"> ▪ Kenntnisse in Python ▪ Kenntnisse in angewandter Mathematik ▪ Grundlagenkenntnisse in Neuronalen Netzen 	
Literatur	<ul style="list-style-type: none"> ▪ Louis G. Birta, Gilbert Arbez: Modelling and Simulation, Exploring Dynamic Systems Behaviour, Springer 2019 ▪ Richard S. Sutton, Andrew G. Barto: Reinforcement Learning, MIT Press, 2020 ▪ Maxim Lapan: Deep Reinforcement Learning Hands-On, Packt 2018 	
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig	
Modulverantwortliche(r)	Prof. Dr. C. Lürig	
Änderungsdatum	03.01.2023	

Software-Architekturen			
Inhalte	<ul style="list-style-type: none"> ▪ Architekturmuster (z.B. Microservices) ▪ Entwurf von Architekturen ▪ Dokumentation von Architekturen ▪ Bewertung von Architekturen ▪ Model-Driven-Architecture™ (MDA) ▪ Enterprise Integration Patterns 		
Lernergebnisse	Die Studierenden können <ul style="list-style-type: none"> ▪ konkrete Architekturen analysieren und verstehen, ▪ passende Architekturen und Architekturmuster auf Problemstellungen anwenden, ▪ Architekturen beschreiben und entwerfen, ▪ Analyse- und Entwurfsmuster anwenden, ▪ spezifische Verfahren zum modellbasierten Architekturentwurf anwenden, und ▪ klassische n-tier und moderne Architekturen verstehen und beurteilen. 		
Lehrform	<input checked="" type="checkbox"/> Vorlesung		
	<input checked="" type="checkbox"/> Übung		
	<input type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Objektorientierte Programmierung - Einführung“, „Software-Entwurf“ und „Grundlagen des Anforderungsmanagements“		
Literatur	<ul style="list-style-type: none"> ▪ Nick Rozanski, Eoin Woods: Software Systems Architecture. 2. Auflage, Addison-Wesley Professional, 2011. ▪ Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice. 3. Auflage, Addison-Wesley Longman, 2012. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input checked="" type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz, Prof. Dr. G. Rock		
Modulverantwortliche(r)	Prof. Dr. C. Schmitz		
Änderungsdatum	04.01.2023		

Software-Qualitätsmanagement		
Inhalte	<ul style="list-style-type: none"> ▪ Einführung und Überblick ▪ Qualitätssicherung ▪ Manuelle Prüfmethode ▪ Verbesserung der Prozessqualität ▪ Produktqualität - Komponenten <ul style="list-style-type: none"> ▪ Testende Verfahren ▪ Verifizierende Verfahren ▪ Analysierende Verfahren ▪ Produktqualität - Systeme <ul style="list-style-type: none"> ▪ Integrationstest ▪ System- und Abnahmetest ▪ Konfigurations- und Änderungsmanagement 	
Lernergebnisse	<p>Die Studierenden können</p> <ul style="list-style-type: none"> ▪ Qualitätsbegriffe definieren und einordnen, ▪ die Prinzipien der Software-Qualitätssicherung erklären und begründen, ▪ (Code-)Inspektionen durchführen, ▪ kontrollflussorientierte und datenflussorientierte Testverfahren einsetzen, ▪ die Konzepte der Verifikation und des symbolischen Testens verwenden und gegen testende Verfahren abgrenzen, ▪ für einfache Beispiele Integrations- und Abnahmetests durchführen, ▪ Testwerkzeuge einsetzen, sowie ▪ Werkzeuge und Verfahren des Konfigurationsmanagements einsetzen. 	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Keine	
Literatur	<ul style="list-style-type: none"> ▪ Roman, Adam: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus: Test Techniques and Sample Mock Exams. Springer, 1. Auflage, 2018. ▪ Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest – Testmanagement - Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB-Standard. dpunkt Verlag, 4. Auflage, 2014. ▪ Liggesmeyer, Peter: Software-Qualität. Spektrum Verlag, 2. Auflage, 2009. 	
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. G. Rock	
Modulverantwortliche(r)	Prof. Dr. G. Rock	
Änderungsdatum	03.02.2023	

Statistics and Learning Theory		
Inhalte	Statistics and statistical learning theory constitute the theoretical foundation of machine learning and data science. The following content is covered in this module: <ul style="list-style-type: none"> ▪ Hypothesis tests: Binomial test, Chi-square test, t-test, ANOVA, Neyman-Pearson lemma, multi hypothesis testing, decision theory ▪ Estimation theory: MLE, MAP, Fisher information + Cramer-Rao lower bound, applications of exponential families, full-Bayesian approach, ▪ Statistical learning theory: Error decomposition and bias-variance tradeoff, PAC Learning bounds, growth functions, VC-dimension, Rademacher complexity, algorithmic stability, ensemble methods Hinweis: Die Vorlesung wird in Englisch gehalten.	
Lernergebnisse	Students learn fundamental principles and deeper concepts, methods and techniques in statistics and statistical learning theory. Students can independently use primary literature, especially in English, to acquire advanced scientific knowledge and competences. They also learn to evaluate the limitations of AI technologies and data science methods and to critically assess related safety and security issues. Students can analyze and structure complex problems, develop their own ideas and implement solutions by means of adequate methods and tools. In this module, students are prepared for further scientific qualifications.	
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt	
Empfohlene Voraussetzungen	Gute Englisch-Kenntnisse	
Literatur	<ul style="list-style-type: none"> ▪ Henze, Norbert: Stochastik für Einsteiger. Springer, 13. Auflage [2021]. ▪ L. Wasserman: All of Statistics. Springer, 2003. ▪ M. Hardt, B. Recht: Patterns, Predictions, and Actions: Foundations of Machine Learning. Princeton University Press, 2022. 	
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen	
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input checked="" type="checkbox"/> PF <input type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig	
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit
	6	60 Stunden
		Selbststudium
		120 Stunden
Lehrende(r)	Prof. Dr. H.-P. Beise	
Modulverantwortliche(r)	Prof. Dr. H.-P. Beise	
Änderungsdatum	25.01.2023	

Ubiquitous Computing			
Inhalte	Die folgenden Themen werden als Aspekte von Systemen im Bereich „Ubiquitous Computing“ betrachtet: <ul style="list-style-type: none"> ▪ Beispielsysteme aus Forschung und Anwendung ▪ Erhebung, Verwendung und Verteilung von Kontextinformationen ▪ „Location“ als besonderer Kontext ▪ Anpassung von Anwendungen an Benutzer und Situation ▪ Alternative Ein- und Ausgabe-, sowie Interaktionsmöglichkeiten ▪ Aufbau und Durchführung von Benutzerstudien ▪ Umgang und Unterstützung von Privacy als Kernproblematik von UbiComp Systemen ▪ Drahtlose Datenkommunikation und Ad-hoc Protokolle ▪ Spracheingabe und Sprachausgabe ▪ Zukünftige Entwicklungen Weiterhin werden wissenschaftliche Papiere aus den relevanten wissenschaftlichen Konferenzen und Journalen diskutiert, um die Themen der Vorlesung an aktuellen Forschungsergebnissen zu spiegeln.		
Lernergebnisse	Studierende haben Kenntnis vom Bereich „Ubiquitous Computing“, sowie den zugehörigen Anwendungen, Konzepten, Verfahrensweisen und aktuellen Forschungsthemen. Sie analysieren Situationen aus dem Blickwinkel des Forschungsgebietes und sind in der Lage aus den Beobachtungen und unter Verwendung der gängigen Methoden und Vorgehensweisen des Gebiets ein UbiComp-System zu konzipieren, zu erstellen und zu evaluieren. Sie sind in der Lage ihre Ergebnisse als wissenschaftliches Papier zu formulieren, im Stil, wie er im Forschungsgebiet üblich ist.		
Lehrform	<input type="checkbox"/> Vorlesung <input type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Web-Technologien“. Die Veranstaltung ist teilnahmebeschränkt. Bei Überbuchung werden die Plätze durch einen Einstufungstest vergeben.		
Literatur	<ul style="list-style-type: none"> ▪ John Krumm : Ubiquitous Computing Fundamentals. Taylor & Francis Ltd, 2009. ▪ Wissenschaftliche Papiere der Proceedings der Konferenzreihe UbiComp, bzw. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) Journal werden im Rahmen der Übung vorgestellt und diskutiert. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende[r]	Prof. Dr. G. Schneider		
Modulverantwortliche[r]	Prof. Dr. G. Schneider		
Änderungsdatum	22.11.2022		

Verlässliche Echtzeitsysteme			
Inhalte	<ul style="list-style-type: none"> ▪ 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 haben die Prinzipien verlässlicher Echtzeitsysteme und die Besonderheiten bei deren Entwicklung kennen gelernt. Sie sind in der Lage, in interdisziplinären Teams bei der Entwicklung von sicherheitsrelevanten Echtzeitsystemen mitzuwirken.		
Lehrform	<input type="checkbox"/> Vorlesung		
	<input type="checkbox"/> Übung		
	<input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht		
	<input checked="" type="checkbox"/> Labor		
	<input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Technische Informatik“		
Literatur	<ul style="list-style-type: none"> ▪ ISO 26262 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avizienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013. 		
Studienleistung	<input type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	<input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input type="checkbox"/> Mündliche Prüfung		
	<input type="checkbox"/> Klausur		
	<input type="checkbox"/> Prüfung am PC		
	<input type="checkbox"/> Hausarbeit (ggf. mit Präsentation)		
	<input checked="" type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science	<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF	
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider		
Modulverantwortliche(r)	Prof. Dr. J. Schneider		
Änderungsdatum	06.02.2023		

Verteilte Systeme			
Inhalte	<p>In diesem Modul werden wichtige Konzepte verteilter Systeme anhand „klassischer“ Originalartikel besprochen. Im Vorlesungsteil werden die wesentlichen Inhalte einer oder mehrerer verwandter Arbeiten vom Dozenten vorgestellt. In den Übungen beantworten die Studierenden Fragen zu den in der Vorlesung präsentierten Sachverhalten, erarbeiten anhand der Literatur Themen, die in der Vorlesung nicht oder nur oberflächlich behandelt wurden, und erstellen dafür eine Präsentation. Ferner implementieren sie einige der behandelten Algorithmen.</p> <p>Die behandelten Themen sind:</p> <ul style="list-style-type: none"> ▪ Einleitung ▪ Verteilte Graph-Algorithmen ▪ Peer-to-Peer-Systeme (CAN, Chord, Pastry/Tapestry, P-Grid) ▪ Terminierungserkennung (Termination Detection) und Abfallsammlung (Garbage Collection) ▪ Verklemmungserkennung ▪ Logische und reale Zeit ▪ Nachrichtenreihenfolge ▪ Konsensfindung und Koordination ▪ Globale Zustände ▪ Globale Prädikate ▪ Transaktionen, insbesondere Nebenläufigkeitskontrolle (2-Phasen-Sperren mit Verklemmungsbehandlung, Zeitstempel-Verfahren, optimistische Verfahren) ▪ Datenreplikation 		
Lernergebnisse	<ul style="list-style-type: none"> ▪ Die Studierenden können grundlegende Probleme verteilter Systeme erläutern und dafür grundlegende Lösungsansätze darstellen (s. Inhalte). ▪ Die Studierenden können sich mit wissenschaftlichen Artikeln (in der Regel in Englisch) aus Zeitschriften und Konferenzen auseinandersetzen, wesentliche Informationen daraus extrahieren und diese für eine Präsentation in gut verständlicher Form aufbereiten. Dabei diskutieren sie die Inhalte kritisch und gehen auf ihre Verständnisprobleme bzw. aus ihrer Sicht entdeckter Ungereimtheiten in den Artikeln ein. ▪ Die Studierenden können verteilte Algorithmen, die in wissenschaftlichen Artikeln in Pseudocode beschrieben werden, in einer Programmiersprache und -umgebung ihrer Wahl implementieren. 		
Lehrform	<input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input checked="" type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ Im Wesentlichen: Mehrere Originalartikel (werden in der Vorlesung bekannt gegeben). Dazu als Hintergrundinformation: ▪ A. Tanenbaum, M. Van Steen: Distributed Systems – Principles and Paradigms, 2nd Edition, 2007. ▪ G. Coulouris, J. Dollimore, T. Kindberg, G. Blair: Distributed Systems – Concepts and Design, 6th Edition, 2013. ▪ A. Kshemkalyani, M. Singhal: Distributed Computing – Principles, Algorithms, and Systems, 2011. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung <input type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. R. Oechsle		
Modulverantwortliche(r)	Prof. Dr. R. Oechsle		
Änderungsdatum	22.11.2022		

Visuelle Navigation			
Inhalte	<p>SLAM – Frontend, Backend, Mapping:</p> <ul style="list-style-type: none"> ▪ Einführung in das Themengebiet ▪ Repräsentationen und Transformationen ▪ Lie-Gruppen und Lie-Algebren ▪ Kameramodelle und Bildentstehung ▪ Konvexe Optimierung (Einführung) ▪ Visuelle Odometrie I (Invariante Merkmale, Epipolar-Geometrie und Registrierung) ▪ Visuelle Odometrie II (Optischer Fluss und Direkte Methoden) ▪ Filter und Optimierung I (Bündelblockausgleich und graphenbasierte Optimierung) ▪ Filter und Optimierung II (Effiziente Löser und Posen-Graph-Optimierung) ▪ Identifikation von Schleifen in Trajektorien ▪ (Dichte) Rekonstruktion aus mehreren Ansichten ▪ Visuelle Stereo-Odometrie und weitere Verfahren <p>Die Unterlagen sind in englischer Sprache verfasst, die Unterrichtssprache ist Deutsch.</p>		
Lernergebnisse	<p>Die Studierenden beherrschen die mathematischen Grundlagen für den Entwurf eines SLAM-basierten Navigationssystems und verfügen über die praktischen Fähigkeiten selbstständig ein ausgewähltes SLAM-Modell in Grundzügen zu implementieren (engl. Self Localization and Mapping).</p> <p>SLAM-basierte Navigation ermöglicht es einem mobilen Roboter durch Erkundung seiner Umwelt mit visuellen Sensoren selbstständig ein Umgebungsmodell aufzubauen und dabei die zurückgelegte Trajektorie zu bestimmen. Die Analyse und Entwicklung eines derartigen Systems werden von den Studierenden beherrscht.</p> <p>Ausgehend von einem in der Vorlesung vorgestellten Systementwurfs werden im Rahmen der Vorlesung verschiedene Konzepte und Methoden diskutiert die anschließend im Rahmen der Übungen anhand ausgewählter Aufgaben implementiert werden. Damit ist der Erwerb der praktischen Fähigkeiten nach erfolgreichem Abschluss der Übungen sichergestellt.</p> <p>Visuelle Navigation stellt nach Einschätzung vieler Experten aktuell eines der wichtigsten und grundlegendsten Herausforderungen der Robotikforschung dar und wird zukünftig spannende und herausfordernde Arbeitsplätze im Bereich der Forschung und Entwicklung im industriellen und akademischen Umfeld schaffen (bspw. im beim autonomen Fahren).</p>		
Lehrform	<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input type="checkbox"/> Seminar/Seminaristischer Unterricht <input type="checkbox"/> Labor <input type="checkbox"/> Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	<ul style="list-style-type: none"> ▪ Yi Ma et al.: An Invitation to 3-D Vision: From Images to Geometric Models. 1. Auflage (2003), Springer. ▪ Richard Hartley & Andrew Zisserman: Multiple View Geometry in computer vision. 2. Auflage (2004), Cambridge University Press. ▪ Xiang Gao & Tao Thang: Introduction to Visual SLAM: From Theory to Practice. 2021, Springer. ▪ Wolfgang Förstner & Bernhard P. Wrobel: Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction. 2018, Springer. 		
Studienleistung	<input checked="" type="checkbox"/> Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation) <input type="checkbox"/> Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation) <input type="checkbox"/> Bestehen von Leistungsstandkontrollen		
Prüfungsform	<input checked="" type="checkbox"/> Mündliche Prüfung (nur bei geringer Teilnehmerzahl) <input checked="" type="checkbox"/> Klausur <input type="checkbox"/> Prüfung am PC <input type="checkbox"/> Hausarbeit (ggf. mit Präsentation) <input type="checkbox"/> Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Künstliche Intelligenz und Data Science		<input type="checkbox"/> PF <input checked="" type="checkbox"/> WPF
Angebot	<input type="checkbox"/> Sommersemester <input checked="" type="checkbox"/> Wintersemester <input type="checkbox"/> Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Graf		
Modulverantwortliche(r)	Prof. Dr. J. Graf		
Änderungsdatum	12.01.2023		