



**HOCHSCHULE
COBURG** | **FAKULTÄT**
Elektrotechnik und Informatik

Modulhandbuch

BACHELORSTUDIENGANG VISUAL COMPUTING (VC)
GÜLTIG FÜR STUDIENANFÄNGER AB 01.10.2025

Studienverlauf Bachelor Visual Computing für SPO ab Wintersemester 2025/26

1. Fachsemester	2. Fachsemester	3. Fachsemester	4. Fachsemester	5. Fachsemester	6. Fachsemester	7. Fachsemester
Analysis (7 6)	Diskrete Mathematik (5 4)	Stochastik (5 4)	Mathematik für Visual Computing (5 4)	Industrie- praktikum (22)	WPF VC (5 4)	WPF VC (5 4)
Grundlagen der Informatik (7 6)	Grundlagen des Visual Computing (7 6)	Computergrafik 1 (5 4)	Computergrafik 2 (5 4)		WPF VC (5 4)	WPF VC (5 4)
		Bildverarbeitung 1 (5 4)	Bildverarbeitung 2 (5 4)		WPF IF (5 4)	WPF IF (5 4)
	Rechner- architekturen (7 6)	Web- technologien (7 6)	Software Engineering (5 4)		Künstliche Intelligenz 1 (5 4)	Künstliche Intelligenz 2 (5 4)
Programmieren 2 (7 6)		Fortgeschrittene Programmierung (5 4)	Mensch- Maschine Interaktion 1 (5 4)		Mensch- Maschine Interaktion 2 (5 4)	
		Englisch 1 (2 2)	Englisch 2 (2 2)		Algorithmen & Datenstrukturen (5 4)	Laborpraktikum (2 2)
WPF-SQF (2 2)	WPF-SQF (2 2)		WIA (3 2)		Praxisseminar (2 2)	

VC: Visual Computing
IF: Informatik
WIA: Wissenschaftliches und Interdisziplinäres Arbeiten
WPF: Wahlpflichtfach
SQF: Schlüsselqualifikation

Zahlen in Klammern: ECTS | SWS
ECTS: European Credit Transfer System
SWS: Semesterwochenstunden
SPO: Studien- und Prüfungsordnung

Stand: SPO vom 17.01.2025 Hochschule Coburg

Informatik	Praxis
Visual Computing	Interdisziplinär

Vorbemerkungen

Ein ECTS-Leistungspunkt nach dem „European Credit and Accumulation Transfer System“ entspricht einer Arbeitsbelastung von 30 Stunden pro Semester.

Die Erläuterungen zu den formalen Zulassungsvoraussetzungen für die einzelnen Module finden Sie in der Studien- und Prüfungsordnung (SPO) des Studiengangs.

Studienziel:

Ziel des Bachelorstudiums ist die Vermittlung der Befähigung zur selbständigen Anwendung wissenschaftlicher Erkenntnisse und Methoden im Bereich Visual Computing, wobei fachbezogene und fachübergreifende Qualifikationsziele verbunden werden, um die Persönlichkeitsentwicklung der Studierenden zu fördern. Visual Computing umfasst alle Vorgänge, bei denen Bilder von Computern erzeugt sowie be- oder verarbeitet werden. Dazu gehören vor allem die Bildsynthese mit Modellierung, Visualisierung und Rendering, die Bildanalyse mit Bildverarbeitung und maschinellem Sehen, die Mensch-Maschine-Interaktion mit virtueller/erweiterter Realität, Benutzerschnittstellendesign und Wahrnehmung, sowie die Künstliche Intelligenz mit Mustererkennung und maschinellem Lernen. Im Visual Computing werden visuelle Informationen mittels Methoden der Mathematik, der Informatik, den Ingenieurwissenschaften und aus Design und Ergonomie verarbeitet. Das Studium soll dazu befähigen, sich rasch in eines der zahlreichen Anwendungsgebiete einzuarbeiten, sich neue Gebiete zu erschließen und sich selbständig weiterzubilden. Der Bachelorabschluss befähigt insbesondere zur Übernahme anwendungsorientierter Fach- und Führungsaufgaben im Bereich des Visual Computing, vor allem in Industrie, Medizintechnik, Spieleentwicklung, neue Medien und Forschung. Die Absolventinnen und Absolventen überblicken, neben den fachlichen Aspekten, auch die fachübergreifenden und gesellschaftlichen Zusammenhänge in den behandelten Fachgebieten und sind in der Lage, tiefergehende wissenschaftliche Methoden und Erkenntnisse anzuwenden, um selbständig relevante Problemstellungen und Aufgaben erkennen und erfolgreich bearbeiten zu können. Dabei sind sie sich ihrer besonderen gesellschaftlichen und individuellen Verantwortung bewusst und handeln entsprechend. Der erfolgreiche Abschluss soll es ermöglichen, das Studium in einem Masterstudiengang national oder international fortzusetzen.

Bitte beachten:

Im Modulhandbuch werden alle Module aufgeführt, für welche im jeweiligen Semester Prüfungen angeboten werden, dabei müssen sie nicht zwingend in diesem Semester gelehrt werden.

Gefährdungsbeurteilung nach §10 Mutterschutzgesetz:

Für jedes Modul existiert eine anlassunabhängige Gefährdungsbeurteilung gemäß §§ 10ff Mutterschutzgesetz (MuSchG). Danach werden die Module nach

grün = „wählbar ohne Einschränkungen“, die Lehrveranstaltung (LV) ist unbedenklich

gelb = „wählbar mit Einschränkungen, individuelle Absprache nötig“, die Teilnahme an der LV bedarf einer Überprüfung im Einzelfall

rot = „nicht im Sinne des MuSchG studierbar“, die Teilnahme an der LV ist für die schwangere Studentin nicht zulässig beurteilt.

Die einzelnen Gefährdungsbeurteilungen finden Sie in den entsprechenden Laboren.

Zentrale Anlaufstelle für eine Beratung schwangerer oder stillender Studentinnen ist das Familienbüro der Hochschule Coburg.

	Gefährdung Mutterschutz		
	Grün	Gelb	Rot
1. Erster Studienabschnitt – theoretische Studiensemester 1 und 2	x		
1.1 Allgemeine Informatik Pflichtmodule	x		
Analysis	x		
Diskrete Mathematik	x		
Grundlagen der Informatik	x		
Programmieren 1	x		
Programmieren 2	x		
Rechnerarchitekturen			
Webtechnologien	x		
1.2 Visual Computing Pflichtmodule			
Grundlagen des Visual Computing	x		
1.3 Schlüsselqualifikationen	x		
Englisch (GER B2) 1	x		
Englisch (GER B2) 2	x		
Wahlpflichtmodul Schlüsselqualifikation: Siehe Fächer des Studium Generale			
2. Erster Studienabschnitt – theoretische Studiensemester 3 und 4			
2.1.1 Fachwissenschaftliche Informatik Pflichtmodule	x		
Algorithmen und Datenstruktur			
Fortgeschrittene Programmierung	x		
2.1.2 Fachwissenschaftliche Visual Computing Pflichtmodule			
Computergrafik 1	x		
Computergrafik 2	x		
Bildverarbeitung 1	x		
Bildverarbeitung 2	x		
Künstliche Intelligenz 1	x		
Künstliche Intelligenz 2	x		
Mathematik für Visual Computing	x		
Mensch-Maschine-Interaktion 1	x		
Mensch-Maschine-Interaktion 2			
Visual-Computing-Seminar	x		
2.2 Schlüsselqualifikationen			
Wissenschaftliches und interdisziplinäres Arbeiten			
3. Dritter Studienabschnitt – theoretische Studiensemester 5 und 7			
3.1 Praktisches Studiensemester			
Industriepraktikum	x		
Praxisbegleitende Lehrveranstaltungen	x		
Praxisseminar	x		
3.2 Fachwissenschaftliche Pflichtmodule			
Künstliche Intelligenz 2	x		
Mensch-Maschine-Interaktion 2	x		

	Gefährdung Mutterschutz		
	Grün	Gelb	Rot
3.3 Fachwissenschaftliche Wahlpflichtmodule Visual Computing			
Bildanalyse und Bildsynthese	x		
3D Visualisierung	x		
Evaluierung von Benutzerschnittstellen	x		
Geometrieverarbeitung	x		
GPU Image Synthesis	x		
Grundlagen Game Design	x		
Grundlagen der Datenkompression	x		
Grundlagen des Maschinellen Lernens	x		
Internet of Things			
3.4 Fachwissenschaftliche Wahlpflichtmodule Informatik			
Betriebssysteme	x		
Cloud Computing	x		
Computernetze	x		
Defensive Security Mechanisms			
Digital Hardware Design	x		
Eingebettete Betriebssysteme	x		
Embeded Softwareentwicklung	x		
E-Entrepreneurship	x		
Fortgeschrittene Webtechnologien			
Grundlagen der Wirtschaftsinformatik	x		
IT-Sicherheit	x		
Rechnerarchitekturen	x		
Robotik		x	
SAP-Systeme – Schnittstellen und ABAP-Programmierung	x		
Secure Software Engineering	x		
Shell und Prozesse	x		
Softwarearchitekturen und Testen	x		
Software-Anforderungen und -Modellierung	x		
Software Engineering	x		
Softwareentwurf in der Automatisierungstechnik	x		
Stochastik	x		
Verteilte Systeme	x		
3.4 Abschlussarbeit			
Bachelorseminar	x		
Bachelorarbeit	x		

Inhaltsverzeichnis

1. Erster Studienabschnitt – theoretische Studiensemester 1 und 2	7
1.1.1 Fachwissenschaftliche Pflichtmodule Informatik.....	7
Analysis	7
Diskrete Mathematik	9
Grundlagen der Informatik	11
Programmieren 1	13
Programmieren 2	15
Rechnerarchitekturen	17
Webtechnologien.....	19
1.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing	22
Grundlagen des Visual Computing.....	22
1.2 Schlüsselqualifikationen (SQF).....	24
Englisch (GER B2) 1	24
Englisch (GER B2) 2	26
Wahlpflichtmodule Schlüsselqualifikationen (SQF).....	28
2. Erster Studienabschnitt – theoretische Studiensemester 3 und 4	29
2.1.1 Fachwissenschaftliche Pflichtmodule Informatik.....	29
Algorithmen und Datenstrukturen	29
Fortgeschrittene Programmierung.....	32
Software Engineering	35
Stochastik	37
2.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing	39
Bildverarbeitung 1.....	39
Bildverarbeitung 2.....	41
Computergrafik 1	43
Computergrafik 2	45
Künstliche Intelligenz 1	47
Mathematik für Visual Computing	49
Mensch-Maschine-Interaktion 1	51
2.2 Schlüsselqualifikationen.....	53
Wissenschaftliches und interdisziplinäres Arbeiten.....	53
3. Dritter Studienabschnitt – Studiensemester 5 bis 7	55
3.1 Praktisches Studiensemester	55
Industriepraktikum	55
Praxisbegleitende Lehrveranstaltungen	57

Praxisseminar	58
3.2 Fachwissenschaftliches Pflichtmodul	60
Mensch-Maschine-Interaktion 2	60
Künstliche Intelligenz 2	62
3.3 Fachwissenschaftliche Wahlpflichtmodule Visual Computing	65
Bildanalyse und Bildsynthese	65
3D Visualisierung	67
Evaluierung von Benutzerschnittstellen	69
GPU Image Synthesis	71
Geometrieverarbeitung	73
Grundlagen Game Design.....	75
Grundlagen des Maschinellen Lernens	77
Grundlagen der Datenkompression	79
Internet of Things.....	81
3.4 Fachwissenschaftliche Wahlpflichtmodule Informatik.....	84
Betriebssysteme	84
Cloud Computing.....	86
Communication Systems.....	89
Computernetze.....	91
Defensive Security Mechanisms.....	93
Digital Hardware Design	96
Eingebettete Betriebssysteme	98
Embedded Softwareentwicklung	101
E-Entrepreneurship	105
Fortgeschrittene Webtechnologien	106
Grundlagen der Wirtschaftsinformatik.....	110
IT-Sicherheit.....	113
Robotik	115
Secure Software Engineering.....	117
Shell und Prozesse	119
Software-Anforderungen und -Modellierung.....	122
Software-Architekturen und -Testen.....	124
Softwareentwurf in der Automatisierungstechnik	126
Verteilte Systeme	129
3.5 Abschlussarbeit	131
Bachelorarbeit.....	131
Bachelorseminar	133

1. Erster Studienabschnitt – theoretische Studiensemester 1 und 2

1.1.1 Fachwissenschaftliche Pflichtmodule Informatik

Modulbezeichnung	Analysis
Kürzel	Ana
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	90 h Präsenz (Seminaristischer Unterricht mit integrierten Übungen) 120 h Eigenarbeit (40 h Nachbereitung des Lehrstoffs, 30 h Bearbeitung von Übungsaufgaben, 50 h Prüfungsvorbereitung)
Fachsemester	1
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Ada Bäumner
Dozent(in)	Prof. Dr. Ada Bäumner
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	-
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	Studierende sollen wesentliche Grundlagen der Analysis bis hin zur Differentialrechnung kennen und anwenden können.
Lehrinhalte	Logik, Mengenlehre, Vollständige Induktion, Kombinatorik, rationale und reelle Zahlen, komplexe Zahlen, Folgen und Grenzwerte, Reihen, Funktionen und Stetigkeit, Differenzierbarkeit, Sätze der Differenzialrechnung Extremwerte, Taylorentwicklung
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (120 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Tafel, Skript

Literatur	<p>T. Arens et al., „Mathematik“, Spektrum, Heidelberg, 2008</p> <p>G. Teschl, S. Teschl, „Mathematik für Informatiker“, Band 1 und 2, Springer Spektrum Berlin, Heidelberg, 2013</p> <p>W. Struckmann, D. Wätjen, „Mathematik für Informatiker“, Springer Vieweg Berlin, Heidelberg, 2016</p> <p>R. Berghammer, „Mathematik für Informatiker“, Springer Vieweg Wiesbaden, 2014</p> <p>E. Weitz, „Konkrete Mathematik (nicht nur) für Informatiker“, Springer Spektrum Wiesbaden, 2018</p> <p>O. Forster, „Analysis 1“, Vieweg, Wiesbaden, 2004</p>
-----------	---

Modulbezeichnung	Diskrete Mathematik
Kürzel	DMth
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 h Präsenz (Seminaristischer Unterricht mit integrierten Übungen) 90 h Eigenarbeit (30 h Nachbereitung des Lehrstoffs, 20 h Bearbeitung von Übungsaufgaben, 40 h Prüfungsvorbereitung)
Fachsemester	2
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Roman Rischke
Dozent(in)	Prof. Dr. Roman Rischke
Sprache	Deutsch
Nutzung in anderen Studiengängen	Modulstudium
Zulassungsvoraussetzungen	-
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	Studierende sollen wesentliche Grundlagen der linearen Algebra und der diskreten Mathematik mit entsprechenden <u>grundlegenden Algorithmen kennen und anwenden können.</u>
Lehrinhalte	Vektorräume, Matrizen und lineare Abbildungen, lineare Gleichungssysteme, Eigenwerte/-vektoren, lineare Optimierung, elementare Zahlentheorie, algebraische Strukturen (Gruppen, Ringe, Körper), Kryptologie und RSA
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (120 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, E-Learning-Medien
Literatur	<ul style="list-style-type: none"> Aigner: Diskrete Mathematik. 6. Auflage, Springer Vieweg, 2006.

	<ul style="list-style-type: none"> • Fischer & Springborn: Lineare Algebra: Eine Einführung für Studienanfänger. 19. Auflage, Springer, 2020. • Matthes: Algebra, Kryptologie und Kodierungstheorie. Fachbuchverlag Leipzig, 2003. • Steger: Diskrete Strukturen. Band 1: Kombinatorik, Graphentheorie, Algebra, 2. Auflage, 2007. • Strang: Lineare Algebra. Springer, 2003. • Teschl & Teschl: Mathematik für Informatiker. Band 1: Diskrete Mathematik und Lineare Algebra, 4. Auflage, Springer Vieweg, 2013.
--	--

Modulbezeichnung	Grundlagen der Informatik
Kürzel	GI
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	90 h Präsenz (Seminaristischer Unterricht: 60 h, Übung: 30 h) 120 h Eigenarbeit (Nachbereitung Seminaristischer Unterricht: 40 h, Übung: 30 h, Prüfungsvorbereitung: 50 h)
Fachsemester	1
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr.-Ing. habil. Tilo Strutz
Dozent(in)	Prof. Dr.-Ing. habil. Tilo Strutz
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Nutzung in anderen Studiengängen	Studiengang Informatik: Pflichtmodul
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	keine
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: <ul style="list-style-type: none"> Studierende sollen die Funktionsweise von Computern und von Grundelementen moderner Programmiersprachen kennen. Studierende sollen einfache Problemstellungen in eine algorithmische Lösung umsetzen können. Studierende sollen Aufgaben, theoretische Grundlagen und grundsätzliche Funktionsweise von Compilern kennen und verstehen. Persönliche-Soziale Kompetenzen: <ul style="list-style-type: none"> Studierende aktivieren ihre Kompetenzen im Bereich des Selbst- und Zeitmanagements.
Lehrinhalte	Einführung <ul style="list-style-type: none"> Historie der Informatik, Daten und Information Zahlendarstellung, Boolesche Algebra Aufbau und Funktionsweise von Rechnern <ul style="list-style-type: none"> Von-Neumann-Rechner Hardware (Hauptplatine, Peripherie, CPU, Busse) Leistungsmerkmale, Einsatzbereiche

	<ul style="list-style-type: none"> • Registermaschine <p>Konzepte von Programmiersprachen</p> <ul style="list-style-type: none"> • Maschinencode, Assembler-, Hochsprache • Datentypen • Prozedurale Abstraktion • Input/Output, Kontrollstrukturen • Pseudocode, Strukto-, Flussdiagramm <p>Komplexität von Algorithmen</p> <p>Datenstrukturen + Suchen</p> <ul style="list-style-type: none"> • Felder, Warteschlange, Stack, Liste, Graph, Baum <p>Algorithmenentwurf</p> <ul style="list-style-type: none"> • Entwurfstechniken, Rekursion, Sortieralgorithmen <p>Automatentheorie</p> <ul style="list-style-type: none"> • DEA, Mealy- und Moore-Automaten • Akzeptierte Sprache • Minimale und Kellerautomaten • Turing-Maschine <p>Informations- und Codierungstheorie</p> <ul style="list-style-type: none"> • Informationsquellen, Informationsgehalt, Entropie • Shannon-Fano- und Huffman-Codierung • Fehlererkennung, Paritätsbit, • Hamming-Distanz, Hamming-Code <p>Softwareentwicklung</p> <ul style="list-style-type: none"> • Vorgehensmodelle, Werkzeuge • Compiler, Linker, Interpreter, Debugger
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overheadprojektor
Literatur	<p>H. Herold et al: Grundlagen der Informatik, 4. Auflage, Pearson, München</p> <p>H.-P. Gumm, M. Sommer: Einführung in die Informatik, Oldenbourg Verlag, München / Wien, 10. Auflage, 2012</p> <p>H. Ernst: Grundkurs Informatik, Vieweg, Braunschweig / Wiesbaden, 4. Auflage, 2008 oder neuer</p>

Modulbezeichnung	Programmieren 1
Kürzel	Prog1
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none"> • 60 h Präsenz (30 h Seminaristischer Unterricht, 30 h Übung) • 90 h Eigenarbeit (30h Vor- und Nachbereitung des Lehrstoffs, 30h Lösung von Übungsaufgaben, 30h Prüfungsvorbereitung)
Fachsemester	1
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Volkhard Pfeiffer
Dozent(in)	Prof. Volkhard Pfeiffer
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	keine
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Studierende sollen</p> <ul style="list-style-type: none"> • die zentralen Konzepte von Programmiersprachen (Datentypen, Variablen, Prozeduren, Kontrollstrukturen, Zeiger, Programmausführung) kennen, verstehen und auf Problemstellungen anwenden können, • die Grundlagen der objektorientierten Programmierung kennen, verstehen und auf Problemstellungen anwenden können. <p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage:</p>

	<ul style="list-style-type: none"> • in Kleingruppen zu kommunizieren und einzelne Programmieren-Übungsaufgaben in 2-er Teams zu bearbeiten und Lösungsvarianten zu reflektieren, • Programmier-Lösungen vor einer größeren Gruppe im Plenum vorzustellen und zu diskutieren, • Zeitressourcen einzuteilen und die notwendige Ausdauer bei der Bearbeitung von Aufgaben, Übungen mit Abgabeterminen und einer 2-er Gruppen-Projektaufgabe nach Zeitplan zu trainieren.
Lehrinhalte	<ul style="list-style-type: none"> • Einführung • Datentypen und Ausdrücke • Kontrollstrukturen • Arrays und Zeiger • Prozedurale Programmierung • Rekursion • Objektorientierte Programmierung – Teil 1 • Strings • Exception Handling
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overheadprojektor, E-Learning Medien
Literatur	<p>Ullenboom, Christian "Java ist auch eine Insel" Galileo Computing jeweils in der neusten Auflage</p> <p>Krüger, Guido "Handbuch der Java Programmierung" Addison Wesley jeweils in der neusten Auflage</p> <p>Kathy, Sierra; Bates, Bert; „Java von Kopf bis Fuß“ O'Reilly jeweils in der neusten Auflage</p> <p>Schiedermeier R. "Programmieren mit Java" Pearson Studium jeweils in der neusten Auflage</p>

Modulbezeichnung	Programmieren 2
Kürzel	Prog2
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	210 h, davon <ul style="list-style-type: none"> • 90 h Präsenz (30 h Seminaristischer Unterricht, 60 h Übung) • 120 h Eigenarbeit (30 h Vor- und Nachbereitung des Lehrstoffs, 60 h Lösung von Übungsaufgaben, 30h Prüfungsvorbereitung)
Fachsemester	2
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Volkhard Pfeiffer
Dozent(in)	Prof. Volkhard Pfeiffer
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Nutzung in anderen Studiengängen	Betriebswirtschaft – Schwerpunkt Wirtschaftsinformatik
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Grundlagen der imperativen Programmierung
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Studierende sollen</p> <ul style="list-style-type: none"> • weiterführende Konzepte der objektorientierten Programmierung kennen, verstehen und auf Problemstellungen anwenden können, • die Grundlagen der Programmierung von Bedienoberflächen und Ein/Ausgabe-Handling kennen, verstehen und auf Problemstellungen anwenden können, • verschiedene Datenstrukturen kennen, verstehen und anwenden können, • die Grundlagen der funktionalen Programmierung im Rahmen einer objekt-funktionalen Sprache kennen, verstehen und anwenden können.

	<p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • in Kleingruppen zu kommunizieren und einzelne Programmieren-Übungsaufgaben in 2-er Teams zu bearbeiten und Lösungsvarianten zu reflektieren, • komplexere Programmier-Lösungen vor einer größeren Gruppe im Plenum vorzustellen und zu diskutieren, • Zeitressourcen einzuteilen und die notwendige Ausdauer bei der Bearbeitung von Aufgaben, Übungen mit Abgabeterminen und einer 2-er Gruppen-Projektaufgabe nach Zeitplan zu trainieren.
Lehrinhalte	<ul style="list-style-type: none"> • Objektorientierte Programmierung - Vertiefung • Collection Datenstrukturen • Utility Klassen • Generics • Funktionale Programmierung • Lambda und Streams • I/O - Ein- und Ausgabe
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overheadprojektor, E-Learning Medien
Literatur	<p>Ullenboom, Christian "Java ist auch eine Insel" Galileo Computing jeweils in der neusten Auflage</p> <p>Krüger, Guido "Handbuch der Java Programmierung" Addison Wesley jeweils in der neusten Auflage</p> <p>Kathy, Sierra; Bates, Bert; „Java von Kopf bis Fuß“ O'Reilly jeweils in der neusten Auflage</p> <p>Schiedermeier R. "Programmieren mit Java" Pearson Studium jeweils in der neusten Auflage</p>

Modulbezeichnung	Rechnerarchitekturen
Kürzel	Ra
Lehrform / SWS	Seminaristischer Unterricht, Übung / 6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	210 h, davon <ul style="list-style-type: none"> • 90 h Präsenz (60 h Seminaristischer Unterricht, 30 h Übungen) • 120 h Eigenarbeit (60 h Nachbereitung des Lehrstoffs, 30 h Bearbeitung von Übungsaufgaben, 30 h Prüfungsvorbereitung)
Fachsemester	1
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Keine
Qualifikationsziele / Kompetenzen	<p>1. Studierende sollen die Fähigkeit erlangen, Rechnerkonzepte in Hard- und Software als Gesamtsystem zu durchdringen.</p> <p>2. Durch Verständnis von logischen Funktionen sowie von elektronischen Verknüpfungsgliedern können Studierende Funktionen in digitale Hardware umzusetzen.</p> <p>3. Studierende verstehen Codierungen, Zahlendarstellungen und grundlegende Algorithmen der Datenverarbeitung und können angepasste Rechensysteme entwerfen.</p>
Lehrinhalte	<p>Axiome der Booleschen Algebra, Normalformen (DNF, KNF)</p> <p>Minimierung von kombinatorischen Schaltungen mit Karnaugh-Veitch, Quine McCluskey</p> <p>Digitale Hardware: MOS-Transistor, Grundgatter der Digitaltechnik, Komplexgatter</p> <p>Standardschaltnetze: Codierer, Decodierer, Multiplexer, Komparatoren, Addierer, Carry Look Ahead</p>

	<p>Zahlendarstellung und Arithmetikfunktionen: Zweierkomplement, Festkomma, Gleitkomma</p> <p>Bitspeicher, Register: SR-Flipflop, D-Latch, D-Flipflop</p> <p>Halbleiterspeicher: DRAM, SRAM, PROM, Flash</p> <p>Architektur von Prozessorsystemen: CPU: ALU, Registerbank, Steuerwerk, Funktionsregister</p> <p>Ablaufsteuerung mit Befehlsverarbeitung, Interrupt, Stack, Pipelining</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Tafel, Beamer
Literatur	<p>Becker, Drechsler, Molitor: Technische Informatik (Pearson Studium)</p> <p>Tanenbaum / Goodman: Computerarchitektur (Pearson Studium)</p> <p>Obermann / Schelp: Rechneraufbau und Rechnerstrukturen (Oldenbourg)</p> <p>Floyd: Digital Fundamentals (Prentice Hall)</p>

Modulbezeichnung	Webtechnologien
Kürzel	Wt
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	<p>210 h, davon</p> <ul style="list-style-type: none"> • 90 h Präsenz (60 h Seminaristischer Unterricht, 30 h Praktikum) • 120 h Eigenarbeit (40 h Vor- und Nachbereitung des Lehrstoffs, 40 h Lösung von Übungsaufgaben, 40 h Prüfungsvorbereitung)
Fachsemester	2
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Prof. Dr. Jürgen Terpin, Prof. Dr. Dieter Wißmann
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Nutzung in anderen Studiengängen	Bachelor Betriebswirtschaft, Bachelor Visual Computing
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Kenntnisse in einer objektorientierten Programmiersprache
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden sollen ein Verständnis für das Zusammenspiel der Konzepte des Internet und des World Wide Web entwickeln. • Sie sollen die relevanten Techniken der Clientseite im Web (Browser) beherrschen lernen, d.h. sie sollen statische Webseiten und Webseiten mit dynamischem Inhalt implementieren können. • Sie sollen die Fähigkeit erlangen, Webseiten konform zu den aktuellen Standards von HTML, JavaScript und CSS zu erstellen. • Sie sollen lernen, die nicht-funktionalen Aspekte bei der Gestaltung von Webauftritten wie Design, Zielgerät und Sicherheit zu berücksichtigen.

	<p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Innerhalb des Praktikums, aber auch im Rahmen des seminaristischen Unterrichts werden Arbeitsaufgaben in Kleingruppen (2-3 Personen) bearbeitet. Die gemeinsame Bearbeitung erfolgt dabei teilweise auch außerhalb der Hochschule (in Präsenz oder online). • Die Studierenden erwerben bzw. verbessern personale Kompetenzen indem sie beispielsweise innerhalb des Gruppenarbeitskontextes gemeinsam <ul style="list-style-type: none"> ○ die zur Zielerreichung notwendigen Abläufe, Ressourcen etc. organisieren/koordinieren, ○ die Gesamtaufgabe in Teilaufgaben zerlegen und diese auf die Gruppenmitglieder verteilen, ○ sich gegenseitig informieren, abstimmen etc. und entsprechende (digitale) Werkzeuge hierfür einsetzen, ○ Konflikte möglichst eigenständig auflösen, ○ komplexe Sachverhalte in der Gruppe diskutieren und strukturieren sowie zielgerichtet und adressatenbezogen im Plenum darstellen/präsentieren.
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen des Internets und des World Wide Webs <ul style="list-style-type: none"> ○ HTTP-Protokoll ○ Architektur eines Browsers ○ Zusammenspiel Browser und Webserver • Technologien auf der Client-Seite (Browser) <ul style="list-style-type: none"> ○ HTML und XHTML ○ Cascading Stylesheets (CSS) ○ JavaScript <ul style="list-style-type: none"> ▪ prozedurale Konzepte ▪ objektorientierte Konzepte ▪ Serialisierung mit JSON ○ API-Konzepte für Webseiten: DOM und BOM ○ AJAX ○ Sicherheitsaspekte bei Webseiten ○ Clientseitige Frameworks (z.B. jQuery) ○ Webseiten für mobile Geräte ○ Responsive Web Design
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Overhead, Tafel, E-Learning Medien
Literatur	<ul style="list-style-type: none"> • Ackermann, Philip (2025): JavaScript. Das umfassende Handbuch, 4. Aufl., Rheinwerk Computing, Bonn.

	<ul style="list-style-type: none"> • Müller, P. (2024): Einstieg in HTML und CSS – Barrierefreie und responsive Webseiten programmieren und gestalten mit HTML und CSS – ganz ohne Vorwissen, 3. Aufl., Rheinwerk Computing, Bonn. • Wolf, J. (2023): HTML und CSS: Das umfassende Handbuch, 5. Aufl., Rheinwerk Computing, Bonn. • Internet- und HTML-Spezifikationen siehe IETF http://www.ietf.org , W3C http://www.w3.org sowie WHATWG https://whatwg.org/
--	---

1.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing

Modulbezeichnung	Grundlagen des Visual Computing
Kürzel	GdVC
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	90 h Präsenz (Seminaristischer Unterricht: 60 h, Übung: 30 h) 120 h Eigenarbeit (Nachbereitung Seminaristischer Unterricht: 40 h, Übung: 30 h, Prüfungsvorbereitung: 50 h)
Fachsemester	2
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Stephan Streuber
Dozent(in)	Prof. Dr. Stephan Streuber
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	keine
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • Grundlegende Konzepte und Begriffe des Visual Computing kennen • Grundlegende Verfahren zur Erfassung, Übertragung, Bearbeitung, Repräsentation, Ausgabe und Speicherung von Bildern kennen, verstehen und anwenden können • Fundamentale Methoden des Visual Computing kennen, verstehen und anwenden • Einfache Visual Computing der Bildmanipulation und Bilderzeugung Problemstellungen algorithmisch und praktisch umsetzen können <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p>

	<ul style="list-style-type: none"> • die Vorteile des gemeinschaftlichen Erarbeitens von Lösungsansätzen erkennen und ausnutzen • Lösungsvorschläge von Programmier- und Theorieaufgaben vor einer Gruppe von Studierenden präsentieren und diskutieren • die Auswirkungen von Visual Computing auf die Gesellschaft, Wirtschaft, Politik und Umwelt verstehen, verstehen, reflektieren und selbstständig erfassen
Lehrinhalte	<ul style="list-style-type: none"> • Das menschliche visuelle System • Einführung in <i>Processing</i> • Farbrepräsentation • Bildrepräsentationen • Speicherung sowie ein- und Ausgabe von Bildern • Diskrete Funktionen • Zahldarstellungen für Visual Computing (Fließkomma, Festkomma) • Hardware für Visual Computing • Grundlegende Techniken der Computergrafik • Grundlegende Bildverarbeitungstechniken • Abtastung, Quantisierung • Entropie, Codierung, Kompression • Ethische Aspekte des Visual Computing und Technikfolgenabschätzungen
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel
Literatur	<p>Deussen, Ninegelen: Programmieren lernen mit Computergrafik: Eine Einführung mit Java und Processing, Springer 2018</p> <p>Angel, Shreiner, Interactive Computer Graphics: A Top-Down Approach with WebGL, 8th edition, Pearson, 2020</p> <p>Birchfield, Image Processing and Analysis, Cengage Learning, 2018</p>

1.2 Schlüsselqualifikationen (SQF)

Modulbezeichnung	Englisch (GER B2) 1
Kürzel	Eng1
Lehrform / SWS	2 SWS
Leistungspunkte	2 ECTS
Arbeitsaufwand	30 h Präsenz, 30 h Eigenarbeit
Fachsemester	1
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Richard Fry, MCLFS
Dozent(in)	Richard Fry, MCLFS
Sprache	Englisch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	empfohlen: Vorkenntnisse der Zielsprache GER B1
Qualifikationsziele / Kompetenzen	<p>Fachkompetenz</p> <ul style="list-style-type: none"> erweiterte aktive und passive Sprachkompetenzen (Sprechen, Schreiben, Hörverstehen, Lesen) mindestens auf der Sprachkompetenzstufe B2 fachspezifischer Schwerpunkt: Fachvokabular, Korrespondenz berufsspezifischer Schwerpunkt: Gesprächsführung, Vorstellungsgespräche <p>Methodenkompetenz</p> <ul style="list-style-type: none"> Erwerb von Lernstrategien, die zum autonomen Lernen befähigen; bestimmte Aufgabenstellungen ermöglichen eine Reflexion über die angewandten Strategien <p>Interkulturelle Kompetenz</p> <ul style="list-style-type: none"> Verwendung der adäquaten Sprache (z.B. Register, Höflichkeitsformen) in interkulturellen Interaktionen in beruflichen und gesellschaftlichen Situationen landeskundliche Kenntnisse englischsprachiger Länder

	<p>Lernkompetenz</p> <ul style="list-style-type: none"> • Selbstlernkompetenzen verstärkt durch das <i>Blended Learning</i> Konzept
Lehrinhalte	<ul style="list-style-type: none"> • wechselnde technische Themen (z.B. Anwendungsprogramme, Betriebssysteme, Rechnerarchitektur, elektronische Datenspeicherungssysteme) • beruflicher Schriftverkehr: Emails, formale Korrespondenz • technisches Schreiben: Berichterstattung, Prozessablauf • Bewerbungsprozess: Lebenslauf, Bewerbungsschreiben, Vorstellungsgespräch
Endnotenbildende Studien- / Prüfungsleistungen	studienbegleitende schriftliche Prüfung (60 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Visualizer
Literatur	Skript

Modulbezeichnung	Englisch (GER B2) 2
Kürzel	Eng2
Lehrform / SWS	2 SWS
Leistungspunkte	2 ECTS
Arbeitsaufwand	30 h Präsenz, 30 h Eigenarbeit
Fachsemester	2
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	R. Fry, MCLFS
Dozent(in)	R. Fry, MCLFS
Sprache	Englisch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	empfohlen: Vorkenntnisse der Zielsprache GER B1
Qualifikationsziele / Kompetenzen	<p>Fachkompetenz</p> <ul style="list-style-type: none"> erweiterte aktive und passive Sprachkompetenzen (Sprechen, Schreiben, Hörverstehen, Lesen) mindestens auf der B2 Sprachkompetenzstufe fachspezifischer Schwerpunkt: Fachvokabular, Korrespondenz berufsspezifischer Schwerpunkt: Gesprächsführung, Vorstellungsgespräche <p>Methodenkompetenz</p> <ul style="list-style-type: none"> Erwerb von Lernstrategien, die zum autonomen Lernen befähigen; bestimmte Aufgabenstellungen ermöglichen eine Reflexion über die angewandten Strategien <p>Interkulturelle Kompetenz</p> <ul style="list-style-type: none"> Verwendung der adäquaten Sprache (z.B. Register, Höflichkeitsformen) in interkulturellen Interaktionen in beruflichen und gesellschaftlichen Situationen landeskundliche Kenntnisse englischsprachiger Länder <p>Lernkompetenz</p>

	<ul style="list-style-type: none"> • Selbstlernkompetenzen verstärkt durch das <i>Blended Learning</i> Konzept
Lehrinhalte	<ul style="list-style-type: none"> • wechselnde technische Themen (z.B. Anwendungsprogramme, Betriebssysteme, Rechnerarchitektur, elektronische Datenspeicherungssysteme) • beruflicher Schriftverkehr: Emails, formale Korrespondenz • technisches Schreiben: Berichterstattung, Prozessablauf • Bewerbungsprozess: Lebenslauf, Bewerbungsschreiben, Vorstellungsgespräch
Endnotenbildende Studien- / Prüfungsleistungen	studienbegleitende schriftliche Prüfung (60 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Visualizer
Literatur	Skript

Wahlpflichtmodule Schlüsselqualifikationen (SQF)

Die Studium Generale Module, die im ersten Semester als Schlüsselqualifikation von Informatik Studierenden belegt werden sollten finden Sie unter:

<https://mycampus.hs-coburg.de/de/service-offer/verantwortung-und-gesellschaft-gesundheit>

Falls Sie sich in **Mathematik** noch etwas unsicher fühlen, empfehlen wir jedoch im ersten Semester den Einführungskurs „**Einstieg in die Mathematik**“ als Schlüsselqualifikationsfach zu besuchen. Sie finden den Kurs unter dieser Rubrik des Studium Generale:

<https://mycampus.hs-coburg.de/de/service-offer/naturwissenschaft-technik>

Anmeldung / Hinweise

Die Anmeldung/Einschreibung erfolgt im Anmeldezeitraum über

Primuss (Eule mit Doktorhut in den Apps am rechten Seitenrand) -> Meine Prüfungen -> Meine Platzvergabe (Einschreibe-/Losverfahren) -> Studium Generale (zuerst den Infotext lesen, dann) -> Zur Anmeldung -> Kurse in die rechte Spalte verschieben

Achtung: Melden sie sich unbedingt im Primuss von Kursen ab die Sie nicht belegen wollen!

Sie blockieren sonst Plätze auf die andere dann nicht nachfolgen können.

Bei **Fragen** zu den Kursen sowie den Anmeldemodalitäten wenden Sie sich bitte an tanja.bayer@hs-coburg.de

Im zweiten Semester kann das SQF frei aus den angebotenen Studium Generale Modulen und den Sprachen gewählt werden!

<https://mycampus.hs-coburg.de/de/interdisciplinary>

2. Erster Studienabschnitt – theoretische Studiensemester 3 und 4

2.1.1 Fachwissenschaftliche Pflichtmodule Informatik

Modulbezeichnung	Algorithmen und Datenstrukturen
Kürzel	ADs
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none">• 60 h Präsenz (Seminaristischer Unterricht: 30 h, Übung: 30 h)• 90 h Eigenarbeit (Vor- und Nachbereitung des Lehrstoffs: 30 h, Lösung der Übungsaufgaben: 30 h, Prüfungsvorbereitung: 30 h)
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Prof. Dr. Florian Mittag
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik B.Sc.
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen der imperativen Programmierung
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: Studierende sollen <ul style="list-style-type: none">• Spezifikationstechniken von Datenstrukturen und Algorithmen kennen, verstehen und anwenden können• Algorithmenanalyse hinsichtlich Komplexität, Speicherbedarf etc. kennen, verstehen und anwenden können• geeignete Datenstrukturen sowie Techniken zum Algorithmenentwurf kennen, verstehen und auf nicht-triviale Probleme anwenden können

	<ul style="list-style-type: none"> • bekannte Algorithmen aus verschiedensten Anwendungsgebieten kennen, verstehen und anwenden können <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • selbstständig weitere Lernquellen (z.B. Visualisierungstools für Algorithmen) recherchieren und in Vor- und Nachbereitung der Veranstaltung integrieren können • die vorgestellten Inhalte untereinander diskutieren und kritisch hinterfragen können
Lehrinhalte	<p>Grundlagen</p> <ul style="list-style-type: none"> • Algorithmenbegriffe • programmiersprachliche Konstrukte zur Algorithmenspezifikation • ausgewählte Algorithmenprobleme und Klassifikation von Algorithmen <p>Algorithmenanalyse</p> <ul style="list-style-type: none"> • Mathematische Grundlagen • Komplexitätsklassen und Laufzeitberechnungen <p>Abstrakte Datentypen und deren Implementierung</p> <ul style="list-style-type: none"> • axiomatische und programmiersprachliche Spezifikation • grundlegende Datenstrukturen (Liste, Stack, Queue etc.) <p>Fortgeschrittene Sortieralgorithmen</p> <ul style="list-style-type: none"> • in-memory und external Sortieralgorithmen <p>Bäume</p> <ul style="list-style-type: none"> • Grundlagen • Binärbäume, Mehrwegbäume • Ausgeglichene Bäume, binäre Suchbäume sowie weitere Baumarten <p>Stringsuche</p> <ul style="list-style-type: none"> • Grundlagen • Knuth-Morris-Pratt-Algorithmus
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer und Tafel/Whiteboard, E-Learning Medien
Literatur	Lang, H.W.: „Algorithmen in Java“, Oldenbourg Verlag jeweils in der neusten Auflage

	<p>Ottmann, T.; Widmayer, P.; „Algorithmen und Datenstrukturen“, Spektrum Verlag jeweils in der neusten Auflage</p> <p>Weiss, M.A.; „Data structures & algorithms Analysis in JAVA“, Addison Wesley jeweils in der neusten Auflage</p> <p>Weiss, M.A.; „Data Structures and Problem Solving Using Java“, Addison Wesley jeweils in der neusten Auflage</p>
--	--

Modulbezeichnung	Fortgeschrittene Programmierung
Kürzel	FProg
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzstudium: 60 h (35 h Seminaristischer Unterricht, 25 h Übungen) Eigenstudium: 90 h (bzw. 25 h Teilnahme an freiwilliger Laborübung + 65 h Eigenstudium)
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Carolin Helbig
Dozent(in)	Prof. Dr. Carolin Helbig
Sprache	Deutsch
Nutzung in anderen Studiengängen	AU, EL und EN (in Semester 7)
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Fundierte Kenntnisse der imperativen Programmierung, etwa aus Programmieren 1 und 2
Qualifikationsziele / Kompetenzen	Die Studierenden erhalten Kenntnisse in der Programmierung in den Sprachen Python und C++ auf verschiedenen Betriebssystemplattformen. Damit werden sie in die Lage versetzt, kleinere Skripte und C++-Anwendungen selbst zu erstellen und größere zu verstehen und zu warten. Der Schwerpunkt liegt mit ca. 2/3 der Veranstaltung auf der Sprache C++.
Lehrinhalte	Python: Geschichte von Python, Operatoren und Datentypen, Variablen, Kommentare, Kontrollstrukturen, Funktionen, Module, Dateien, Ausnahmebehandlung, Klassen, Objekte, Vererbung C++: Unterschiede zu C: Die C++-Programmiersprache, Ein- und Ausgabekanäle, Namensräume, Referenzen und Parameterübergabe, Vorgabewerte für Parameter, Dynamische Speicherverwaltung Klassen und Objekte: Klassendeklaration und -definition, Objekte von Klassen, Zugriffsbeschränkungen, Freunde, Zugriffsroutinen, Konstruktoren, Standardkonstruktor,

	<p>Initialisierung mit Listen, Kopierkonstruktor, Typumwandlungskonstruktor, Destruktoren, Inline-Funktionen,</p> <p>Vererbung: Basisklassen und abgeleitete Klassen, Vererbung in C++, Erzeugung von Unterklassenobjekten, Zugriffsbeschränkungen</p> <p>Polymorphismus: Grundprinzip, Virtuelle Methoden, Virtuelle Destruktoren, rein virtuelle Funktionen und abstrakte Klassen</p> <p>Templates: Funktionstemplates, Klassentemplates, Operatoren zur Typumwandlung</p> <p>Die STL: die Containerklassen der C++-Standardbibliothek: Strings, Container, Iteratoren, Algorithmen, Speichermanagement</p> <p>Ausnahmebehandlung (Exceptions)</p> <p>Dateien und Ströme: Ein- und Ausgabe mit Dateien, Positionierung, Ausgabeformatierung</p> <p>Überladen von Operatoren: Operatorfunktionen, Indexoperator, Zuweisungsoperator, Mathematische Operatoren, Ein- und Ausgabeoperator</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten) als computergestützte Präsenzprüfung
Sonstige Leistungsnachweise	-
Medienformen	<p>Beamer und Tafel/Whiteboard,</p> <p>Elektronische Skripten und Arbeitsunterlagen,</p> <p>Gedrucktes ausführliches Skript (ca. 300 Seiten),</p> <p>E-Learning-Umgebung Moodle mit Selbsttests</p>
Literatur	<p>T. Wieland: C++-Entwicklung mit Linux. dpunkt-Verlag, 2004</p> <p>E.-E. Doberkat: Python 3 – Ein Lehr- und Arbeitsbuch. De Gruyter, 2018</p> <p>S. Dörn: Python lernen in abgeschlossenen Lerneinheiten. Springer-Vieweg, 2020</p> <p>C. Schäfer: Schnellstart Python. Springer-Spektrum, 2019</p> <p>B. Stroustrup: Die C++-Programmiersprache. Addison-Wesley, 2000</p> <p>U. Kirch, P. Prinz: C++ Lernen und professional anwenden, mitp, 8. Auflage, 2018</p> <p>B. Stroustrup: Die C++-Programmiersprache. Hanser, 2015</p> <p>U. Breymann: Der C++-Programmierer, Hanser-Verlag, 5. Aufl., 2017</p>

	B. Stroustrup: Eine Tour durch C++, Hanser, 2015 A. Wilms: C++: eine kompakte Einführung, dpunkt-Verlag, 2015
--	--

Modulbezeichnung	Software Engineering
Kürzel	SwE
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 h Präsenz (Seminaristischer Unterricht: 45 h, Praktikum: 15 h) 90 h Eigenarbeit (Seminaristischer Unterricht: 60 h, Praktikum: 30 h)
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Dieter Landes
Dozent(in)	Prof. Dr. Dieter Landes
Sprache	Deutsch
Nutzung in anderen Studiengängen	Bachelor Betriebswirtschaft
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Programmierkenntnisse
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: <ul style="list-style-type: none"> • Studierende sollen grundlegende Konzepte der professionellen Software-Entwicklung kennen und erklären können. • Studierende sollen Techniken und Notationen der professionellen Software-Entwicklung in Rahmen einer nicht-trivialen Anwendungsfragestellung verwenden können.
Lehrinhalte	<ul style="list-style-type: none"> • Einführung <ul style="list-style-type: none"> ◦ Einleitung und Motivation • Vorgehensmodelle <ul style="list-style-type: none"> ◦ Software-Lebenszyklus ◦ Plangetriebene Vorgehensmodelle ◦ Agile Vorgehensmodelle • Anforderungserhebung und Analyse <ul style="list-style-type: none"> ◦ Anforderungen ◦ Aktivitäten bei der Anforderungsanalyse ◦ UML-Modelle in der Systemanalyse ◦ Objektorientierte Analyse

	<ul style="list-style-type: none"> • Design <ul style="list-style-type: none"> o Entwurfskriterien o Standard-Softwarearchitekturen • Softwaretest <ul style="list-style-type: none"> o Testprozess o Testebenen und -arten
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten), praktische Studienarbeit (Gewicht 3:1)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Overhead, Tafel, Softwaremodellierungswerkzeuge
Literatur	<p>I. Sommerville: Software Engineering, Addison-Wesley, Boston, 9. Auflage, 2010</p> <p>B. Oestereich: Analyse und Design mit UML 2.5, Oldenbourg, München, 10. Auflage, 2012</p> <p>C. Rupp: Requirements-Engineering und -Management, Hanser, München, 5. Auflage, 2009</p>

Modulbezeichnung	Stochastik
Kürzel	Sto
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 h Seminaristischer Unterricht 90 h Eigenarbeit (60 h Bearbeitung von Übungsaufgaben, 30 h Prüfungsvorbereitung)
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Roman Rischke
Dozent(in)	Prof. Dr. Roman Rischke
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Kenntnisse aus Analysis und Diskreter Mathematik
Qualifikationsziele / Kompetenzen	Fachkompetenz <ul style="list-style-type: none"> Studierende sollen grundlegende mathematische Denkweisen, Begriffe und Techniken der Stochastik beherrschen Methodenkompetenz <ul style="list-style-type: none"> Studierende sollen zufällige Phänomene mathematisch erfassen und Problemlösungen entwickeln können
Lehrinhalte	<ul style="list-style-type: none"> Grundlagen der Differential- und Integralrechnung Grundlagen der Stochastik: Berechnung von Wahrscheinlichkeiten, diskrete und stetige Verteilungen und deren Kenngrößen Markov-Ketten: Übergangswahrscheinlichkeiten, Charakterisierung von Zuständen Statistik: Punkt- und Intervallschätzungen, Testtheorie
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (120 Minuten)

Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, E-Learning-Medien
Literatur	<ul style="list-style-type: none"> • Henze: Stochastik für Einsteiger: Eine Einführung in die faszinierende Welt des Zufalls. 13. Auflage, Springer Spektrum, 2021. • Hübner: Stochastik. 4. Auflage, Vieweg, 2003. • Papula: Mathematik für Ingenieure und Naturwissenschaftler. Band 1 und 3, Springer Vieweg. • Schickinger & Steger: Diskrete Strukturen 2: Wahrscheinlichkeitstheorie und Statistik. Springer, 2002. • Teschl & Teschl: Mathematik für Informatiker. Band 2: Analysis und Statistik, 3. Auflage, Springer Vieweg, 2014.

2.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing

Modulbezeichnung	Bildverarbeitung 1
Kürzel	BiVa 1
Lehrform / SWS	Vorlesung + Übung / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h 60 h Präsenz, 90 h Eigenarbeit (60 h Vor- und Nachbereitung des Lehrstoffs, 30 h Prüfungsvorbereitung)
Fachsemester	3
Angebotsturnus	Jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr.-Ing. habil. Tilo Strutz
Dozent(in)	Prof. Dr.-Ing. habil. Tilo Strutz
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul (VC)
Nutzung in anderen Studiengängen	Studiengang „Informatik“: Wahlpflichtmodul im 3. Semester
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Höhere Mathematik, Kenntnisse in Programmierung (C, Java oder Matlab)
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Die Studierenden kennen und beherrschen die Grundlagen und Anwendungen der Bildverarbeitung und Computer Vision.</p> <p>Insbesondere sind die Studierenden befähigt, wichtige Verfahren der Bildverarbeitung zu verstehen und anzuwenden. Die Studierenden beherrschen mathematische Grundlagen (lineare Algebra, Elemente der Systemtheorie). Sie sind zudem befähigt, technische Systeme der Bildverarbeitung und Computer Vision umzusetzen. Dazu benötigte grundlegende Methoden können sie erklären und ggf. mit alternativen Methoden vergleichen (z. B. Einsatz von verschiedenen Verfahren zur Berechnung von Bildmerkmalen). Die Studierenden erlernen theoretische und praktische Kompetenzen in der Konzeption, Umsetzung und technischen</p>

	<p>Evaluierung von Software und technischen Systemen der Bildverarbeitung und Computer Vision.</p> <p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage, in Kleingruppen zu kommunizieren, gemeinsam Programmieraufgaben zu bearbeiten und zu präsentieren. Sie bauen dabei ihre Kompetenzen im Bereich des Selbst- und Zeitmanagements aus.</p>
Lehrinhalte	<p>Studierende lernen Werkzeuge und Vorgehensweisen kennen um Systeme der Bildverarbeitung und Computer Vision entwerfen und entwickeln zu können. Lehrinhalte umfassen:</p> <ul style="list-style-type: none"> • Bildaufnahme • Punktoperatoren, • Theorie zweidimensionaler Signale (Korrelationsfunktion, Faltung) • lokale Operatoren (lin. Filter, Rangordner) • Kantendetektion, Eckendetektion • Texturanalyse • Verarbeitung binärer Bilder (morphologische Operatoren) • Charakterisierung von 2D-Objekten
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	Prüfungsvorleistung: Softwareprojekt
Medienformen	Präsentation mit Beamer, Gruppenarbeit, E-Learning Medien
Literatur	<ul style="list-style-type: none"> • Erhardt, A.: Einführung in die Digitale Bildverarbeitung • Burger, W.: Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ • Gonzalez, R.C.: Digital Image Processing • Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.

Modulbezeichnung	Bildverarbeitung 2
Kürzel	BiVa 2
Lehrform / SWS	Vorlesung (2 SWS) + Übung (2 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h 60 h Präsenz, 90 h Eigenarbeit (60 h Vor- und Nachbereitung des Lehrstoffs, 30 h Prüfungsvorbereitung)
Fachsemester	4
Angebotsturnus	Jährlich
Dauer des Moduls	Einsemestrig
Modulverantwortliche(r)	Prof. Dr. habil. Jens Grubert
Dozent(in)	Prof. Dr. habil. Jens Grubert
Sprache	Deutsch
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Bildverarbeitung 1, lineare Algebra
Qualifikationsziele / Kompetenzen	<p>Die Studierenden kennen fortgeschrittene Konzepte und Anwendungen der Bildverarbeitung und Computer Vision.</p> <p>Insbesondere sind die Studierenden befähigt wichtige Verfahren der Bildverarbeitung und Computer Vision zu verstehen und anzuwenden. Die Studierenden beherrschen mathematische Grundlagen (lineare Algebra, lineare und nichtlineare Optimierung). Sie sind zudem befähigt Bildverarbeitungssysteme umzusetzen. Dazu benötigte grundlegende Algorithmen können sie erklären und ggf. mit alternativen Algorithmen vergleichen (z.B. Einsatz von verschiedenen Verfahren zur Berechnung von Bildmerkmalen). Die Studierende erlernen theoretische und praktische Kompetenzen in der Konzeption, Umsetzung und technischen Evaluierung von Bildverarbeitungssystemen.</p>
Lehrinhalte	Studierende lernen Werkzeuge und Vorgehensweisen kennen um fortgeschrittene Systeme der Bildverarbeitung und Computer Vision entwerfen und entwickeln zu können. Lehrinhalte umfassen:

	<ul style="list-style-type: none"> • Projektive Geometrie • Robuste Parameterschätzung • Maschinelles Lernen für Computer Vision • Kamerakalibrierung • 3D Registrierung • 3D Rekonstruktion
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Präsentation mit Beamer, Gruppenarbeit, E-Learning Medien
Literatur	<p>Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.</p> <p>Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.</p> <p>Golub, G. H., & Van Loan, C. F. (2012). Matrix computations (Vol. 3). JHU Press. Press,</p> <p>W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Numerical recipes 3rd edition: The art of scientific computing. Cambridge university press.</p>

Modulbezeichnung	Computergrafik 1
Kürzel	CG1
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60h Präsenz (Seminaristischer Unterricht 30h, Übungen und Praktika 30h) 90h Eigenarbeit (Seminaristischer Unterricht 45h, Übungen und Praktika 45h)
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik (B.Sc.)
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Programmierkenntnisse, Lineare Algebra, Analysis
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • grundlegende Konzepte der Computergrafik kennen und erklären können. • Techniken und Konzepte der Grafikprogrammierung im Rahmen einer nicht-trivialen Anwendungsfragestellung verwenden können <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • Gesellschaftliche und ethnische Auswirkung von Computergrafik auf Gesellschaft, Wirtschaft, Politik und Umwelt • komplexere Aufgabenstellung in Zweier-Gruppen kooperativ lösen und präsentieren • Lösungsvorschläge von Programmier- und Theorieaufgaben vor einer Gruppe von Studierenden präsentieren und diskutieren

Lehrinhalte	<ul style="list-style-type: none"> • C Programmierung für Computergrafik • Mathematik für Computergrafik • Geometrirepräsentationen • Linienrasterung • Dreiecksrasterung • Baryzentrische Interpolation • Graphik-Pipeline • Clipping • Transformationen (Perspektive und Projektionen) • Sichtbarkeitsbestimmung • Schattierung • Lokale Beleuchtungsmodelle •
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, 3D-Grafikprogramme
Literatur	<p>Angel, Shreiner, Interactive Computer Graphics: A Top-Down Approach with WebGL, 8th edition, Pearson, 2020</p> <p>Gortler: Foundations of 3D computer graphics. Cambridge, Mass: MIT Press, 2012</p> <p>Shirely, Marschner: Fundamentals of Computer Graphics, Taylor & Francis Ltd., 2015</p>

Modulbezeichnung	Computergrafik 2
Kürzel	CG2
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60h Präsenz (Seminaristischer Unterricht 30h, Übungen und Praktika 30h) 90h Eigenarbeit (Seminaristischer Unterricht 45h, Übungen und Praktika 45h)
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO (IF) Vorrückensberechtigung nach §5 Abs. 1 SPO (VC)
Inhaltliche Voraussetzungen	Programmierkenntnisse, Höhere Mathematik
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • weiterführende Konzepte der Computergrafik kennen, erklären und umsetzen. • weiterführende Techniken und Konzepte der Grafikprogrammierung im Rahmen einer umfangreiche Anwendungsfragestellung umsetzen • können eigenständige neue Computergrafiktechniken verstehen, erklären und programmiertechnisch umsetzen <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • komplexere Aufgabenstellung in Zweier-Gruppen kooperativ lösen und präsentieren • Gesellschaftliche und ethnische Auswirkung von Computergrafik auf Gesellschaft, Wirtschaft, Politik und Umwelt

	<ul style="list-style-type: none"> • Lösungsvorschläge von Programmier- und Theorieaufgaben vor einer Gruppe von Studierenden präsentieren und diskutieren
Lehrinhalte	<ul style="list-style-type: none"> • Texturdarstellung • GPU Hardware • Programmierung • MIP-Mapping • Partielle Ableitung von multivariaten Funktionen und deren Approximation • Height Maps, Normal Maps • Ray-Tracing • Bounding Volume Hierarchies • Alpha-Blending • Anti-Aliasing Verfahren
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, 3D-Grafikprogramme
Literatur	<p>Gortler: Foundations of 3D computer graphics. Cambridge, Mass: MIT Press, 2012</p> <p>Shirely, Marschner: Fundamentals of Computer Graphics, Taylor & Francis Ltd., 2015</p> <p>Angel, Shreiner, Interactive Computer Graphics A Top-Down Approach with WebGL, Pearson Global Edition</p> <p>Peter Shirley, Trevor David Black, Steve Hollasch: Ray Tracing in One Weekend, 2025</p>

Modulbezeichnung	Künstliche Intelligenz 1
Kürzel	KI1
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	180h
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Prof. Dr. Michael Sammeth
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik B.Sc.
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Grundlagen der imperativen Programmierung
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die grundlegenden Konzepte und Begriffe der Künstlichen Intelligenz kennen. • grundlegende Algorithmen der Künstlichen Intelligenz und Verfahren zur Problemlösung durch Suche kennen, verstehen und anwenden können • die Rahmenbedingungen und relevanten Eigenschaften von Problemstellungen kennen und bestimmen können. • die vorgestellten Algorithmen implementieren und zur Lösung von Problemen einsetzen können <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die gesellschaftlichen Einflüsse und ethischen Aspekte der vorgestellten Verfahren untereinander diskutieren, analysieren und eigenständig bewerten können. • Verfahren der Künstlichen Intelligenz anhand aktueller Entwicklungen und Ereignisse einordnen, analysieren und bewerten können.

Lehrinhalte	<ul style="list-style-type: none"> • Was ist KI? • Intelligente Agenten (symbolische KI) • Problemlösen durch Suche <ul style="list-style-type: none"> o Tiefen- und Breitensuche o A*, RBFS (kürzeste Route) o Suche im kontinuierlichen Suchraum o Suche mit Nicht-Determinismus o Suche mit versteckten Zuständen • Suche in Spielen (mit Gegnern) <ul style="list-style-type: none"> o Alpha-Beta-Pruning o Heuristische Echtzeit-Entscheidungen o Stochastische Spiele
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	Russell, Norvig: Artificial Intelligence: A Modern Approach, Addison Wesley, jeweils in der neusten Auflage

Modulbezeichnung	Mathematik für Visual Computing
Kürzel	MathVC
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Prof. Dr. Florian Mittag
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik Bachelor
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen des Visual Computing
Qualifikationsziele / Kompetenzen	<p>Studierende sollen</p> <ul style="list-style-type: none"> • Koordinatentransformationen und Rotationsrepräsentationen in 2D und 3D kennen, verstehen und anwenden • Optimierungsverfahren kennen, verstehen und anwenden • Elemente und Verfahren der multivariaten Analysis kennen und anwenden • Eigenwerte bestimmen und Singulärwertzerlegungen durchführen können
Lehrinhalte	<ul style="list-style-type: none"> • Koordinatentransformationen • Rotationsrepräsentationen in 3D (Matrixdarstellung, Komplexe Zahlen, Eulersche Winkel, Quaternionen, Rodrigues-Rotation, SO3, Lie-Algebra) • Optimierung <ul style="list-style-type: none"> o Methode der kleinsten Quadrate o Lineare Optimierung, lineare Ausgleichsrechnung o Nicht-lineare Optimierung

	<ul style="list-style-type: none"> • Multivariate Analysis (Kurven, Flächen) • Eigenwerte und Singulärwertzerlegung
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	-

Modulbezeichnung	Mensch-Maschine-Interaktion 1
Kürzel	MMI1
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Carolin Helbig
Dozent(in)	Prof. Dr. Carolin Helbig
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik Bachelor
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Grundlagen des Visual Computing
Qualifikationsziele / Kompetenzen	Studierende kennen die technischen, gestalterischen und menschlichen Grundlagen der Mensch-Maschine-Interaktion. Sie verstehen relevante Rahmenbedingungen menschlicher Wahrnehmung und Motorik und deren Einfluss auf die Gestaltung von Mensch-Maschine-Schnittstellen. Studierende können grundlegende interaktive Systeme gestalten, implementieren und evaluieren.
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen des Menschen (Wahrnehmung und Motorik) • Grundlagen des Rechners (Eingabegeräte, Ausgabegeräte) • Grundlagen der Interaktion (Bedienhandlungen, Interaktionsmodelle, Fehlerarten) • Kognitive Modelle (GOMS, KLM) • Anforderungsanalyse (Aufgaben- und Prozessanalyse, Prozessmodellierung, Nutzeranalyse) • Evaluation (Grundbegriffe, qualitative und quantitative Verfahren) • Deskriptive Statistik • Inferenzstatistik

Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Vorlesungsfolien, Lernmanagementsystem, Videos.
Literatur	<p>Dix, A., Dix, A. J., Finlay, J., Abowd, G. D., & Beale, R. (2003). Human-computer interaction. Pearson Education.</p> <p>Lazar, J., Feng, J. H., & Hochheiser, H. (2017). Research methods in human-computer interaction. Morgan Kaufmann.</p> <p>Norman, D. A. (1988). The psychology of everyday things. Basic books.</p> <p>Rogers, Y., Sharp, H., & Preece, J. (2011). Interaction design: beyond human-computer interaction. John Wiley & Sons.</p> <p>Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: strategies for effective human-computer interaction. Pearson.</p>

2.2 Schlüsselqualifikationen

Modulbezeichnung	Wissenschaftliches und interdisziplinäres Arbeiten
Kürzel	WiA
Lehrform / SWS	2 SWS
Leistungspunkte	3 ECTS
Arbeitsaufwand	90 h, davon <ul style="list-style-type: none"> • 30 h Präsenz (15 h Seminaristischer Unterricht, 15 h Seminar) • 60 h Eigenarbeit (Vortragsvorbereitung, schriftliche Ausarbeitung)
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Alle Professoren der Informatik
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Studierende sollen die Grundregeln wissenschaftlichen Arbeitens, auch über disziplinäre Grenzen hinweg, kennen. • Studierende sollen grundlegende Modelle der Kommunikation kennen und verstehen. • Studierende sollen eine wissenschaftliche Fragestellung unter Beachtung der Regeln wissenschaftlichen Arbeitens und von Kommunikationsmodellen schriftlich ausarbeiten und mündlich präsentieren können.

	<p>Persönliche-Soziale Kompetenzen:</p> <p>Die Studierenden sollen im Rahmen der Erstellung der ersten wissenschaftlichen Arbeit im Studium</p> <ul style="list-style-type: none"> • die Gesamtaufgabe strukturieren (z.B. in Arbeitspakete aufteilen) • den Zeitbedarf für die Arbeitspakete abschätzen und unter Einbeziehung ihrer sonstigen Verpflichtungen einen realistischen Zeitplan mit Meilensteinen erstellen, • Fragen/Probleme präzise formulieren und im Seminar mit der Gesamtgruppe diskutieren, • ihre Arbeit zielgerichtet und adressatenbezogen im Plenum präsentieren und diskutieren, • sich verbal und schriftlich wissenschaftlich ausdrücken und • sich kritisch mit dem eigenen Thema und mit Themen anderer Studierender wissenschaftlich und gesellschaftlich auseinandersetzen und diese reflektieren – auch und gerade über disziplinäre Grenzen hinweg.
Lehrinhalte	<ul style="list-style-type: none"> • Grundregeln wissenschaftlichen Arbeitens <ul style="list-style-type: none"> ◦ Wissenschaftliches Recherchieren ◦ Exzerpieren und Paraphrasieren ◦ Zitierregeln ◦ Wissenschaftlicher Schreibstil • Kommunikationsmodelle <ul style="list-style-type: none"> ◦ Eisbergmodell, 4-Ohren-Modell • Studentische Präsentationen
Endnotenbildende Studien- / Prüfungsleistungen	Hausarbeit und Präsentation
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Overhead, Tafel
Literatur	<p>Je nach Thema sowie</p> <p>H. Balzert, M. Schröder, C. Schäfer: Wissenschaftliches Arbeiten. W3L-Verlag, Dortmund, 2011</p> <p>M. Kornmeier: Wissenschaftlich schreiben leicht gemacht: Für Bachelor, Master und Dissertation, 8. Aufl., utb, Haupt Bern, 2018</p>

3. Dritter Studienabschnitt – Studiensemester 5 bis 7

3.1 Praktisches Studiensemester

Modulbezeichnung	Industriepraktikum
Kürzel	
Lehrform / SWS	0 SWS
Leistungspunkte	22 ECTS
Arbeitsaufwand	660 h Eigenarbeit
Fachsemester	5
Angebotsturnus	halbjährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	-
Sprache	-
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Ziele:</p> <ul style="list-style-type: none">• Studierende kennen typische Abläufe und Aufgabenstellungen im Berufsbild des Informatikers.• Studierende sind in der Lage, ihre im Studium erworbenen Kenntnisse auf die angeleitete Bearbeitung einer typischen Aufgabenstellung in einem Industriebetrieb oder einer öffentlichen Einrichtung anzuwenden. <p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none">• Studierende arbeiten in Teams oder Arbeitsgruppen in einem Industriebetrieb• Studierende lernen die formellen und informellen Abläufe und Strukturen in einem Industriebetrieb kennen

	<ul style="list-style-type: none"> • Studierende lösen vielfältige Kommunikationsaufgaben im Team, gegenüber Vorgesetzten und/oder Kunden • Studierende lernen den Umgang mit und das Lösen von Konflikten sowie Selbstmanagement unter Stressbedingungen (z.B. Deadlines)
Lehrinhalte	Abhängig von der Aufgabenstellung
Endnotenbildende Studien- / Prüfungsleistungen	-
Sonstige Leistungsnachweise	Praktikumszeugnis
Medienformen	-
Literatur	-

Modulbezeichnung	Praxisbegleitende Lehrveranstaltungen
Kürzel	-
Lehrform / SWS	4 SWS
Leistungspunkte	6 ECTS
Arbeitsaufwand	60 h Präsenz (Seminaristischer Unterricht) 90 h Eigenarbeit (Lösung der Übungsaufgaben: 30 h, Vor- und Nachbereitung des Lehrstoffs: 60 h)
Fachsemester	5
Angebotsturnus	halbjährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Wechselnde Dozentinnen bzw. Dozenten
Sprache	-
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	Fachlich-methodische Ziele: <ul style="list-style-type: none"> Studierende kennen und reflektieren ausgewählte fachliche Themengebiete mit besonderer Relevanz für das Industriepraktikum. Studierende erhalten die Gelegenheit, überfachliche Kompetenzen mit besonderer Relevanz für das Industriepraktikum zu trainieren.
Lehrinhalte	Nach Festlegung im Studien- und Prüfungsplan
Endnotenbildende Studien- / Prüfungsleistungen	Anwesenheit
Sonstige Leistungsnachweise	-
Medienformen	-
Literatur	-

Modulbezeichnung	Praxisseminar
Kürzel	-
Lehrform / SWS	2 SWS
Leistungspunkte	2 ECTS
Arbeitsaufwand	30 h Präsenz (Seminar) 30 h Eigenarbeit (Vorbereitung des Seminarvortrags)
Fachsemester	5
Angebotsturnus	halbjährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Dozentinnen und Dozenten der Fakultät Elektrotechnik und Informatik
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Ziele:</p> <ul style="list-style-type: none"> • Der Studierende ist in der Lage, schriftlich und mündlich darzustellen, welche typische Aufgabenstellung in einem Industriebetrieb oder einer öffentlichen Einrichtung im Rahmen des Industriepraktikums bearbeitet wurde, welche Herausforderungen sich dabei stellten und welche Lösungsansätze gewählt wurden. <p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Studierende erarbeiten sich selbstständig einen Zeitplan zur Erstellung des Praxisberichts und der Präsentation • Studierende bewerten und entscheiden, welche Inhalte sie in welcher Form präsentieren, unter Berücksichtigung von Betriebsgeheimnissen, fehlendem Vorwissen bei Zuhörern und zeitlichen Einschränkungen.
Lehrinhalte	Abhängig von der Aufgabenstellung

Endnotenbildende Studien- / Prüfungsleistungen	Praxisbericht und Präsentation
Sonstige Leistungsnachweise	-
Medienformen	Beamer
Literatur	-

3.2 Fachwissenschaftliches Pflichtmodul

Modulbezeichnung	Mensch-Maschine-Interaktion 2
Kürzel	MMI2
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Stephan Streuber
Dozent(in)	Prof. Dr. Stephan Streuber
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Mensch-Maschine-Interaktion 1
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Studierende sollen</p> <ul style="list-style-type: none"> • die technischen, gestalterischen und menschlichen Grundlagen der Mensch-Maschine-Interaktion kennen • relevante Rahmenbedingungen menschlicher Wahrnehmung und Motorik und deren Einfluss auf die Gestaltung von Mensch-Maschine-Schnittstellen verstehen • Benutzerschnittstellen gestalten, implementieren und evaluieren können <p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • Projekte gemeinsam zu planen und im Laufe des Semesters umzusetzen (Projektmanagement)

	<ul style="list-style-type: none"> • Entwürfe und Ergebnisse im Plenum vorzustellen, zu diskutieren, und kritisch zu reflektieren • Entwicklungsschritte zu dokumentieren
Lehrinhalte	<ul style="list-style-type: none"> • Wahrnehmung, Motorik, Kognition • Quantitative Methoden in der MMI • Statistische Tests (CHI2, Regression, T-Tests) • Erweiterte Benutzerschnittstellen • 3D-Interaktion (VR/AR/XR, Orientierung, Navigation, Interaktion, Avatare und virtuelle Agenten) • Biophysiological Schnittstellen • Interaktive Datavisualisierung
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Vorlesungsfolien, Unity3D Entwicklung
Literatur	<p>Preim, B., & Dachzelt, R. (2010). Interaktive systeme: band 1: grundlagen, graphical user interfaces, informationsvisualisierung. Springer-Verlag.</p> <p>Preim, B., & Dachzelt, R. (2015). Interaktive Systeme: Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces. Springer-Verlag.</p> <p>Cacioppo, J. T., Tassinary, L. G., & Berntson, G. (Eds.). (2007). Handbook of psychophysiology. Cambridge university press.</p> <p>Döring, N., & Bortz, J. (2016). Forschungsmethoden und evaluation. Wiesbaden: Springerverlag.</p>

Modulbezeichnung	Künstliche Intelligenz 2
Kürzel	KI 2
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Florian Mittag
Dozent(in)	Prof. Dr. Thomas Wieland
Sprache	Deutsch
Nutzung in anderen Studiengängen	Visual Computing B.Sc.
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen der imperativen Programmierung
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die Architektur wissensbasierter und lernender Agenten verstehen und grundlegende Konzepte symbolischer, probabilistischer und datengetriebener KI einordnen können. • Probleme formal in Aussagen- und Prädikatenlogik modellieren sowie geeignete Inferenzverfahren anwenden und deren Leistungsfähigkeit bewerten können. • Unsicherheit mithilfe der Wahrscheinlichkeitstheorie modellieren und Bayessche Inferenzverfahren zur rationalen Entscheidungsfindung einsetzen können. • probabilistische grafische Modelle (z. B. Bayessche Netze) interpretieren und für einfache Problemstellungen anwenden können. • grundlegende Verfahren des überwachten Lernens (z. B. lineare und logistische Regression, Entscheidungsbäume) verstehen, implementieren und hinsichtlich Modellgüte

	<p>und Generalisierungsfähigkeit evaluieren können.</p> <ul style="list-style-type: none"> • geeignete Evaluationsmetriken auswählen, Ergebnisse interpretieren und Modelle vergleichend analysieren können. <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • ethische, gesellschaftliche und rechtliche Implikationen von KI-Systemen (z. B. Bias, Fairness, Transparenz) analysieren und kritisch reflektieren können. • KI-Verfahren anhand aktueller Anwendungen einordnen, bewerten und argumentativ fundiert diskutieren können. • in Teamarbeit Lösungsansätze entwickeln, präsentieren und konstruktiv reflektieren können. • die Verantwortung bei der Entwicklung intelligenter Systeme erkennen und Prinzipien verantwortungsvoller KI in Projektkontexte übertragen können.
Lehrinhalte	<ul style="list-style-type: none"> • Architektur intelligenter Agenten und rationale Entscheidungsfindung • Aussagenlogik und Prädikatenlogik erster Stufe sowie formale Wissensrepräsentation • Inferenzverfahren (Forward/Backward Chaining, Resolution, Unifikation) • Probabilistisches Schließen und Bayessche Wahrscheinlichkeit • Probabilistische grafische Modelle (Bayessche Netze, Hidden Markov Models) • Grundlagen des überwachten Lernens (lineare und logistische Regression) • Entscheidungsbäume, Random Forests und Modellbewertung • Ethische, gesellschaftliche und rechtliche Aspekte der Künstlichen Intelligenz
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	<ul style="list-style-type: none"> • S.J. Russell, P. Norvig: Artificial Intelligence: A Modern Approach, Pearson Studium, 4. Auflage, 2023.

	<ul style="list-style-type: none"> • W. Ertel: Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung (6. Aufl.). Springer Vieweg Wiesbaden, 2025. • R. Kruse, C. Borgelt et al.: Computational Intelligence. Springer Vieweg, 2015. • S. Raschka, V. Mirjalili: Machine Learning mit Python, MITP, 3. Aufl., 2021 S.H. Vieweg: KI für das Gute, Springer Gabler, 2023.
--	---

3.3 Fachwissenschaftliche Wahlpflichtmodule Visual Computing

Modulbezeichnung	Bildanalyse und Bildsynthese
Kürzel	BaBs
Lehrform / SWS	Programmierprojekt / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h 30 h Präsenz, 120 h Eigenarbeit (10 h Vor- und Nachbereitung des Lehrstoffs, 100 Programmierung+Test, 10 h Prüfungsvorbereitung)
Fachsemester	6
Angebotsturnus	Jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr.-Ing. habil. Tilo Strutz
Dozent(in)	Prof. Dr.-Ing. habil. Tilo Strutz
Sprache	Deutsch
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Grundlagen der Bildverarbeitung, Kenntnisse Programmierung (C, Java oder Matlab)
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Die Studierenden beherrschen Methoden zur Bildanalyse, können diese erklären und in einem praktischen Szenario anwenden.</p> <p>Sie kennen verschiedene Verfahren zu Bildsynthese und haben gelernt diese effektiv einzusetzen.</p> <p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage, in Kleingruppen zu kommunizieren, gemeinsam Programmieraufgaben zu bearbeiten und zu präsentieren. Sie bauen dabei ihre Kompetenzen im Bereich des Selbst- und Zeitmanagements aus.</p>
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen der Bildanalyse <ul style="list-style-type: none"> o Grunddimensionen Farbe und Tiefe o Strukturen, Formen, Objekte o Objektdetektion

	<ul style="list-style-type: none"> o Gesichtsdetektion o Dimension Zeit (Bewegung) • Grundlagen Bildsynthese <ul style="list-style-type: none"> o Modifikation o Kreation o Interaktion o Animation • Erstellen einer Software-Lösung für eine Bildanalyse-Bildsynthese-Anwendung (im Team)
Endnotenbildende Studien- / Prüfungsleistungen	SPA
Sonstige Leistungsnachweise	-
Medienformen	Vorlesungsfolien, Lernmanagementsystem
Literatur	<ul style="list-style-type: none"> • Escriva et al.: Building Computer Vision Projects with OpenCV 4 and C++: Implement complex computer vision algorithms and explore deep learning and face detection • Birchfield: Image Processing and Analysis • Gonzales & Woods: Digital Image Processing, Pearson • https://szeliski.org/Book

Modulbezeichnung	3D Visualisierung
Kürzel	3D-Visual
Lehrform / SWS	4 SWS: Seminaristischer Unterricht (2 SWS), Übung (2 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden, 90h Stunden Eigenarbeit
Fachsemester	7.
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Carolin Helbig
Dozent(in)	Prof. Dr. Carolin Helbig
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	Grundlagen Visual Computing, Programmieren 1 und 2
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die grundlegenden Prinzipien und Methoden der Visualisierung kennen, verstehen und anwenden können. • Algorithmen der Visualisierung kennen, verstehen und einsetzen können. • Visualisierungsmethoden für reale Datensätze auswählen und anwenden. <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • Bewusstsein für Potentiale und Gefahren bei der Visualisierung von Daten erwerben (deskriptive vs. suggestive Datenvisualisierung). • in Gruppen zusammenzuarbeiten • Projekte gemeinsam zu planen und im Laufe des Semesters umzusetzen (Projektmanagement) • Entwürfe und Ergebnisse im Plenum vorzustellen, zu diskutieren, und kritisch zu reflektieren • Entwicklungsschritte zu dokumentieren

Lehrinhalte	<ul style="list-style-type: none"> • Begriffe, Abgrenzungen zu anderen Konzepten • Visualisierungsprozess • Datencharakterisierung • Einflussfaktoren • Techniken für einfache, multifacettierte, räumliche und zeitliche, Volumen- und Strömungsdaten • Interaktion • Umgang mit Visualisierungssoftware
Endnotenbildende Studien- / Prüfungsleistungen	SPA
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	<p>Tamara Munzner. Visualization Analysis and Design. A K Peters Visualization Series, CRC Press, 2014.</p> <p>Heidrun Schumann, Wolfgang Müller : Visualisierung - Grundlagen und allgemeine Methoden . Springer Berlin, Heidelberg , 2013</p> <p>Weitere Literatur wird in der LV bekanntgegeben.</p>

Modulbezeichnung	Evaluierung von Benutzerschnittstellen
Kürzel	EvalB
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenz: 60h, Selbststudium: 120h
Fachsemester	6. oder 7. Fachsemester
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Stephan Streuber
Dozent(in)	Prof. Dr. Stephan Streuber
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	-
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Literaturrecherchen durchführen • Qualitative und quantitative Daten erheben • Experimente und Studien designen, programmieren und durchführen • Komplexe Daten analysieren <p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Projektmanagement • Entwürfe und Ergebnisse im Plenum präsentieren • Feedback erhalten und umsetzen
Lehrinhalte	<ul style="list-style-type: none"> • Arten der Evaluation • Überblick verschieden Methoden der Evaluierung • Experimentelle Forschung und Benutzerstudien <ul style="list-style-type: none"> o Hypothesen und Hypothesen Tests

	<ul style="list-style-type: none"> o Experimentelles Design o Durchführung o Statistische Auswertung o Ethische Aspekte • Praktische Umsetzung einer Benutzerstudie als Gruppenarbeit: <ul style="list-style-type: none"> o Studierende Präsentieren ein aktuelles Paper o Studierende designen und programmieren ein Experiment (Gruppenarbeit) o Durchführung des Experiments o Datenanalyse und Aufbereitung o Präsentation der Ergebnisse
Endnotenbildende Studien- / Prüfungsleistungen	SPA
Sonstige Leistungsnachweise	-
Medienformen	Vorlesungsfolien
Literatur	<p>Jakob Nielsen: Usability Engineering. Morgan Kaufmann, San Francisco 1994.</p> <p>F. Sarodnick, H. Brau: Methoden der Usability Evaluation – Wissenschaftliche Grundlagen und praktische Anwendung. Hans Huber, Bern 2006.</p> <p>Miriam Eberhard-Yom: Usability als Erfolgsfaktor. Cornelsen Verlag Berlin, 2010.</p>

Modulbezeichnung	GPU Image Synthesis
Kürzel	GIms
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60h Präsenz (Seminaristischer Unterricht 30h, Praktikum 30h) 90h Eigenarbeit (Praktikum, Erstellung der praktischen Studienarbeit)
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer
Sprache	Deutsch
Nutzung in anderen Studiengängen	IF B.Sc.
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5
Inhaltliche Voraussetzungen	Programmierkenntnisse, Höhere Mathematik, Computergrafik
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • industrierelevante Techniken der interaktiven Computergrafiken und Bildsynthese verstehen, implementieren, beherrschen und erklären können. • können eigenständige neue interaktive Computergrafiktechniken verstehen, erklären und programmiertechnisch umsetzen • können moderne Graphik APIs verwenden und Probleme der interaktiven Computergrafik mittels dieser APIs lösen <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • komplexere Aufgabenstellung in kleine Gruppen kooperativ lösen und präsentieren • Lösungsvorschläge von Programmier- und Theorieaufgaben vor einer Gruppe von Studierenden präsentieren und diskutieren

Lehrinhalte	<ul style="list-style-type: none"> • Modern Graphics API • Complex Scene Rendering • Render-To-Texture • Post-Processing • Deferred Shading • Advanced Texture Mapping Techniques • Advanced Lighting
Endnotenbildende Studien- / Prüfungsleistungen	SPA
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, 3D-Grafikprogramme
Literatur	<p>Gortler: Foundations of 3D computer graphics. Cambridge, Mass: MIT Press, 2012</p> <p>Shirely, Marschner: Fundamentals of Computer Graphics, Taylor & Francis Ltd., 2015</p> <p>Angel, Shreiner, Interactive Computer Graphics A Top-Down Approach with WebGL, Pearson Global Edition</p> <p>Akeniense-Möller, Haines, Hoffmann, Real-Time Rendering, 3rd Edition, Taylor Francis Ltd</p> <p>Haines, Akeniense-Möller, Ray Tracing Gems: High Quality and Real-Time Rendering with DXR and Other APIs, Apress, 2019</p> <p>Fachaufsätze</p> <p>Graphik-API Dokumentation und Referenz</p>

Modulbezeichnung	Geometrieverarbeitung
Kürzel	Geover
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60h Präsenz (Seminaristischer Unterricht 35h, Übungen 25h) 90h Eigenarbeit (Seminaristischer Unterricht 50h, Übung 40h)
Fachsemester	6 oder 7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO (IF) Vorrückensberechtigung nach §5 Abs. 2 SPO (VC)
Inhaltliche Voraussetzungen	Programmierkenntnisse, Höhere Mathematik
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: <ul style="list-style-type: none"> Studierende sollen grundlegende Konzepte der Geometrischen Modellierung und Computeranimation kennen und erklären können. Studierende sollen Techniken und Konzepte der Geometrischen Modellierung und Computeranimation im Rahmen einer nicht-trivialen Anwendungsfragestellung verwenden können
Lehrinhalte	Bézierkurven, Splineskurven, Datenstrukturen für diskrete Netze, Netzreduktion, Netzglättung, Tensorproduktflächen, Unterteilungsflächen, interpolationsbasierte Animation
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, 3D-Grafikprogramme

Literatur	Parent: Computer Animation Algorithms and Techniques, Third Edition, Morgan Kaufmann, 2012 Farin, Curves and Surface for CAGD, Morgan Kaufmann, 2002
-----------	---

Modulbezeichnung	Grundlagen Game Design
Kürzel	GruGaD
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Stephan Streuber
Dozent(in)	Prof. Dr. Stephan Streuber
Sprache	Englisch
Nutzung in anderen Studiengängen	Bachelor Informatik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Studierende sollen:</p> <ul style="list-style-type: none"> • die Grundstruktur von Spielen kennen • Designelemente identifizieren können • neue Spiele gestalten und umsetzen können • die gesellschaftliche Relevanz von Spielen verstehen • erste praktische Erfahrung in der Spieleprogrammierung mit Hilfe von Game Engines (Unity3D) sammeln <p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • in Gruppen zusammenzuarbeiten • Projekte gemeinsam zu planen und im Laufe des Semesters umzusetzen (Projektmanagement) • Entwürfe und Ergebnisse im Plenum vorzustellen, zu diskutieren, und kritisch zu reflektieren

	<ul style="list-style-type: none"> • Entwicklungsschritte zu dokumentieren • Englische Lehrinhalte zu verstehen
Lehrinhalte	<ul style="list-style-type: none"> • Terms and Classifications • Design Basics and Game Elements • Story Telling and Aesthetics • Game Mechanics and Story Telling • Game Testing and Feedback • Emerging Trends and Technologies
Endnotenbildende Studien- / Prüfungsleistungen	SPA
Sonstige Leistungsnachweise	
Medienformen	Präsentation mit Beamer, Gruppenarbeit, Programmierung
Literatur	<ul style="list-style-type: none"> • Koster, R. (2013). Theory of fun for game design. "O'Reilly Media, Inc." • Schell, J. (2008). The Art of Game Design: A book of lenses. CRC press. • Fullerton, T. (2014). Game design workshop: a playcentric approach to creating innovative games. CRC press. • Rogers, S. (2014). Level Up! The guide to great video game design. John Wiley & Sons.

Modulbezeichnung	Grundlagen des Maschinellen Lernens
Kürzel	GML
Lehrform / SWS	4 SWS: Seminaristischer Unterricht (2 SWS), Übung (2 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden, 90h Stunden Eigenarbeit
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Thomas Wieland
Dozent(in)	Prof. Dr. Thomas Wieland
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	lineare Algebra
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Studierende sollen</p> <ul style="list-style-type: none"> • Daten analysieren, aufbereiten und geeignete Verfahren zur Skalierung, Kodierung und Feature Selection auswählen sowie deren Einfluss auf die Modellgüte bewerten können. • die mathematischen Grundlagen zentraler Verfahren wie Support Vector Machines und neuronaler Netze verstehen und anwenden können. • neuronale Netze vom Perzeptron bis zu CNNs, RNNs und Transformern theoretisch erläutern und praktisch implementieren können. • Backpropagation, Optimierungsverfahren und Regularisierungstechniken nachvollziehen und gezielt einsetzen können. • generative Modelle (Autoencoder, VAE, GAN) konzeptionell einordnen und Anwendungsszenarien ableiten können. • Hyperparameter-Optimierung systematisch durchführen sowie Trainings- und Evaluationsstrategien vergleichen können.

	<ul style="list-style-type: none"> • Reinforcement-Learning-Probleme modellieren und Verfahren wie Q-Learning oder Deep Q-Networks implementieren können. • Frameworks wie PyTorch oder TensorFlow sicher zur praktischen Umsetzung einsetzen können. <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • KI-Werkzeuge reflektiert und wissenschaftlich korrekt einsetzen können. • Grenzen, Risiken und Bias von ML-Modellen erkennen und deren gesellschaftliche sowie ethische Auswirkungen kritisch bewerten können. • komplexe Modelle adressatengerecht dokumentieren, präsentieren und im Team weiterentwickeln können.
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen des Maschinellen Lernens <ul style="list-style-type: none"> ◦ CRISP-DM-Prozess ◦ Datenaufbereitung und Feature Engineering • Überwachtes Lernen <ul style="list-style-type: none"> ◦ Support Vector Machines ◦ Neuronale Netze (Perzeptron, MLP, CNN, RNN) • Generative Modelle: Autoencoder, VAE, GAN • Transformer und LLMs: Attention, Pretraining, Fine-Tuning • Hyperparameter-Optimierung • Reinforcement Learning: MDP, Q-Learning, DQN
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	<ul style="list-style-type: none"> • C.M. Bishop: <i>Pattern Recognition and Machine Learning</i>. Springer-Verlag, 2006 • S. Raschka: <i>Machine Learning mit Python</i>, MITP-Verlag, 3. Auflage, 2021 • J. Frochte: <i>Maschinelles Lernen</i>, Hanser-Verlag, 3. Aufl., 2020 • A. Geron: <i>Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow</i>, O'Reilly-Verlag, 2. Aufl., 2020 • M.P. Deisenroth, A.A. Faisal, C. Soon Ong: <i>Mathematics for Machine Learning</i>, Cambridge University Press, 2020

Modulbezeichnung	Grundlagen der Datenkompression
Kürzel	GDkomp
Lehrform / SWS	Vorlesung + Übung / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h 60 h Präsenz, 90 h Eigenarbeit (60 h Vor- und Nachbereitung des Lehrstoffs, 30 h Prüfungsvorbereitung)
Fachsemester	6 oder 7
Angebotsturnus	Jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr.-Ing. habil. Tilo Strutz
Dozent(in)	Prof. Dr.-Ing. habil. Tilo Strutz
Sprache	Deutsch
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Höhere Mathematik, Kenntnisse Programmierung (C, Java oder Matlab)
Qualifikationsziele / Kompetenzen	Die Studierenden kennen und verstehen die wesentlichen Grundprinzipien und Methoden der Datenkompression und deren Anwendung in modernen Systemen. Sie sind in der Lage, für verschiedene Arten von Daten geeignete Methoden zur Kompression auszuwählen, zu kombinieren, anzuwenden und zu bewerten.
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen der Informationstheorie (Information, Entropie, bedingte und Verbundentropie, Redundanz, Irrelevanz) • Entropiecodierung (Huffman-, Rice-) • Präcodierung (Lauflängen-, Phrasen-, u.a.) • Datenreduktion (Unterabtastung, Quantisierung) • Dekorrelation (Prädiktion, Signaltransformationen) • visuelle Wahrnehmung und Farbe • Standards zur Bildkompression (JPEG, JPEG-LS) • Grundlagen der Bildsequenzkompression
Endnotenbildende Studien- / Prüfungsleistungen	Softwareprojekt (1/6) Schriftliche Prüfung (90 Minuten, 5/6)
Sonstige Leistungsnachweise	-

Medienformen	Vorlesungsfolien, Lernmanagementsystem
Literatur	<ul style="list-style-type: none"> • Strutz: Bilddatenkompression, 5.Auflage oder höher

Modulbezeichnung	Internet of Things
Kürzel	IoT
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzstudium: 60 h Selbststudium/Teamarbeit: 120 h
Fachsemester	6. oder 7. Fachsemester
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Thomas Wieland
Dozent(in)	Prof. Dr. Thomas Wieland
Sprache	Deutsch und Englisch
Zuordnung zum Curriculum	Fachwissenschaftliches Wahlpflichtmodul
Nutzung in anderen Studiengängen	Integriertes Produktdesign
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Netzwerkprotokolle
Qualifikationsziele / Kompetenzen	<p>Fachliche Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden lernen die technischen Grundlagen sowie die Anwendungsmöglichkeiten von Funktechnologien, Lokalisierungsverfahren und Softwareplattformen im Kontext des Internet der Dinge kennen. • Sie analysieren und bewerten Geschäftsmodelle für IoT-Anwendungen und wenden formale Methoden wie den Business Model Canvas und den Technology Readiness Level auf bestehende Produkte an. <p>Methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden konzipieren und realisieren im interdisziplinären Team eine praxisnahe IoT-Anwendung unter Berücksichtigung technischer Anforderungen, Benutzerinteraktion und kreativer Gestaltung. • Sie setzen strukturierte Analyse- und Planungstechniken ein, um technische und

	<p>wirtschaftliche Aspekte eines IoT-Systems systematisch zu erfassen und umzusetzen.</p> <p>Persönlich-soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden bearbeiten fachliche Aufgaben sowohl eigenständig als auch in Gruppen, wobei sie Verantwortung übernehmen und ihre Arbeitsergebnisse gemeinsam abstimmen. • Sie präsentieren ihre technischen und gestalterischen Lösungen adressatengerecht und reflektieren den Arbeitsprozess im Team.
Lehrinhalte	<p>Introduction</p> <p>Business Models</p> <ul style="list-style-type: none"> ○ Business Model Canvas ○ Product as a Service ○ Information Service Providers ○ Business model patterns for digitally charged products ○ Technology readiness level <p>Hardware Platforms</p> <ul style="list-style-type: none"> ○ Basics of wireless communication ○ Hardware platforms for IoT devices ○ Sensors <p>Wireless Communication</p> <ul style="list-style-type: none"> ○ Wireless LAN ○ Bluetooth ○ LPWAN ○ ZigBee <p>Cellular Networks</p> <ul style="list-style-type: none"> ○ 2G / GSM ○ 3G / UMTS ○ 4G / LTE ○ NB-IoT ○ 5G and beyond <p>Sensor Data Processing</p> <ul style="list-style-type: none"> ○ Software platforms for nodes ○ Sensor data forwarding ○ Event detection <p>IoT Platforms: AWS, Azure & Co.</p> <p>Security in IoT environments</p> <p>Positioning</p> <ul style="list-style-type: none"> ○ Satellite navigation ○ Indoor positioning
Endnotenbildende Studien- / Prüfungsleistungen	Portfolio
Sonstige Leistungsnachweise	-
Medienformen	<p>Gruppenarbeiten, Konzeptentwicklung, Programmierung</p> <p>Beamer und Tafel/Whiteboard,</p> <p>Elektronische Skripten und Arbeitsunterlagen</p>

Literatur	<ul style="list-style-type: none"> • Vorwiegend aktuelle Forschungsarbeiten • Bök, Patrick-Benjamin; Noack, Andreas; Müller, Marcel; Behnke, Daniel (2020). Computernetze und Internet of Things. Springer-Verlag. • Cirani, Simone; Ferrari, Gianluigi; Picone, Marco; Veltri, Luca (2018). Internet of Things: Architectures, Protocols, and Standards. Wiley & Sons. • Saleem, T.J. and Chishti, M.A. (2021). Big Data Analytics for the Internet of Things. Wiley & Sons • IOT Analytics: IoT Platforms - The central backbone for the Internet of Things, 2015 • J. Sen: Internet of Things – Technology, Applications and Standardization. Intech Open Book, 2018 • D. Gupta et al. (eds.): Smart Sensors for the Industrial Internet of Things. Springer, 2021 • Gassmann, Oliver; Frankenberger, Karolin; Csik, Michaela (2013): Geschäftsmodelle entwickeln: 55 innovative Konzepte mit dem St. Galler Business Model Navigator, Hanser Verlag
-----------	---

3.4 Fachwissenschaftliche Wahlpflichtmodule Informatik

Modulbezeichnung	Betriebssysteme
Kürzel	Bs
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 h Präsenz (Seminaristischer Unterricht: 45 h, Übung: 15 h) 90 h Eigenarbeit (Seminaristischer Unterricht: 45 h, Übung: 45 h)
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Dieter Wißmann
Dozent(in)	Prof. Dr. Dieter Wißmann
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen der Informatik und der Computertechnik / -architektur
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden sollen Notwendigkeit und Vorteile von Betriebssystemen verstehen sowie einen Überblick über Betriebssystemarchitekturen erhalten. • Sie sollen ein Verständnis für die Problematik der Synchronisation von Prozessen und Threads erlangen und die Fähigkeit erwerben, konzeptionelle Synchronisationslösungen zu erstellen. • Sie sollen Methoden zum Erkennen und Vermeiden von Deadlocksituationen anwenden lernen. • Sie sollen Verständnis erlangen, wie ein Betriebssystem Arbeitsspeicher verwaltet,

	Dateisysteme aufgebaut sind und Massenspeicher angebunden werden.
Lehrinhalte	<ul style="list-style-type: none"> • Betriebssystemarchitekturen • Prozesse, Threads und Scheduling • Synchronisation von Prozessen und Threads <ul style="list-style-type: none"> o Schutzmechanismen o Deadlockerkennung • Arbeitsspeicherverwaltung <ul style="list-style-type: none"> o Adressierungsmodelle, Adressräume o Speicherzuteilungsverfahren • Dateisysteme und -verwaltung • Massenspeicher und Speichermedien • E/A-Systemkonzepte
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overheadprojektor; Elektronisches Skript und Arbeitsunterlagen
Literatur	<p>Glatz E.: Betriebssysteme; dpunkt-Verlag, 2010.</p> <p>Silberschatz A., Galvin P., Gagne G.: Operating System Concepts; 9. Auflage, John Wiley & Sons Inc., 2012.</p> <p>Tanenbaum A.: Moderne Betriebssysteme; 3. Auflage, Pearson Education, 2009.</p>

Modulbezeichnung	Cloud Computing
Kürzel	CloudC
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none"> • 60 h Präsenz (30 h Seminaristischer Unterricht, 30 h Praktikum) • 90 h Eigenarbeit (30 h Vor- und Nachbereitung des Lehrstoffs, 30 h Lösung von Übungsaufgaben, 30 h Prüfungsvorbereitung)
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Prof. Dr. Jürgen Terpin
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden können die Grundzüge der Technologien erklären, welche das Cloud Computing (CC) ermöglichen. • Sie können die unterschiedlichen Nutzungsarten bzw. Bereitstellungsformen des CC unterscheiden. • Sie sind in der Lage den potentiellen Nutzen und die grundsätzlichen Risiken des CC für Organisationen/Unternehmen zu erläutern. • Sie können Nutzen, Risiken und Herausforderungen für vorgegebene Cloud-Anwendungsszenarien einschätzen. • Sie sind in der Lage vorgegebene Basis- IT-Infrastruktur-Szenarien im Hinblick auf eine Migration in die Cloud zu analysieren und die entsprechende (virtuelle) IT-Infrastruktur zur Nutzung von

	<p>Rechenleistung, Datenspeicher etc. in der Cloud aufzubauen.</p> <p>.Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Im Rahmen der praktischen Anteile des Moduls werden regelmäßig Arbeitsaufgaben in Kleingruppen (3-4 Personen) bearbeitet. Die gemeinsame Bearbeitung erfolgt dabei teilweise auch außerhalb der Hochschule (in Präsenz oder online). • Die Studierenden erwerben bzw. verbessern personale Kompetenzen indem sie beispielsweise innerhalb des Gruppenarbeitskontextes gemeinsam <ul style="list-style-type: none"> o die zur Zielerreichung notwendigen Abläufe, Ressourcen etc. organisieren/koordinieren, o die Gesamtaufgabe in Teilaufgaben zerlegen und diese auf die Gruppenmitglieder verteilen, o sich gegenseitig informieren, abstimmen etc. und entsprechende (digitale) Werkzeuge hierfür einsetzen, o Konflikte möglichst eigenständig auflösen, o komplexe Sachverhalte in der Gruppe diskutieren und strukturieren sowie zielgerichtet und adressatenbezogen im Plenum darstellen/präsentieren.
Lehrinhalte	<ul style="list-style-type: none"> • Grundsätzliche CC-Konzepte inkl. zugrundeliegende Technologien. • (Informations-)Sicherheit in der Cloud • Rechtliche Aspekte des CC • Kosten/Abrechnung der Cloud-Nutzung • Aufbau von (virtueller) IT-Infrastruktur in der Cloud, hier: Nutzung der Ressourcen der "Amazon Web Services (AWS) Academy"
Endnotenbildende Studien- / Prüfungsleistungen	Praktische Studienarbeit
Sonstige Leistungsnachweise	
Medienformen	Beamer, Tafel, E-Learning Medien
Literatur	<ul style="list-style-type: none"> • Erl, Thomas; Mahmood, Zaigham; Puttini, Ricardo; Wise-Martinez, Pamela J. (2013): Cloud computing. Concepts, technology & architecture. Vancouver, Upper Saddle River, N.J., Munich: Prentice Hall. • Stender, Daniel (2020): Cloud-Infrastrukturen. Das Handbuch für DevOps-Teams und

	<p>Administratoren. 1. Aufl. Bonn: Rheinwerk Computing.</p> <ul style="list-style-type: none"> • Schulungsunterlagen des Cloud Service Providers, hier: "Amazon Web Services (AWS) Academy
--	---

Modulbezeichnung	Communication Systems
Kürzel	CS
Lehrform / SWS	Seminaristischer Unterricht (2 SWS), Übung und Praktikum (2 SWS) / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzzeit: 60h, Selbststudium: 90h
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Matthias Mörz
Dozent(in)	Prof. Dr. Matthias Mörz
Sprache	deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	
Inhaltliche Voraussetzungen	Grundlagen der Digitaltechnik und der digitalen Signalverarbeitung, Programmierkenntnisse
Qualifikationsziele / Kompetenzen	<p>Nach der Veranstaltung können die Studierenden</p> <ul style="list-style-type: none"> • Wichtige drahtlose Kommunikationssysteme beschreiben und charakterisieren • Einfache Bussysteme beschreiben und anwenden • Einfache Hardwareplattformen charakterisieren, auswählen und anwenden • Geeignete Funkmodule für drahtlose Übertragungssysteme auswählen und anwenden • Einfache drahtlose Kommunikationssysteme aufbauen und in Betrieb nehmen • Sensor- und Aktorknoten über einfache Hardwareplattformen auslesen bzw. ansteuern und die Daten per Funk zu übertragen
Lehrinhalte	<ul style="list-style-type: none"> • Überblick über wichtige drahtlose Kommunikationssysteme • Einfache Bussysteme wie I²C, SPI und UART

Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung 90 Minuten und praktische Studienarbeit
Sonstige Leistungsnachweise	keine
Medienformen	Beamer, Präsentationsfolien, Tafel/Whiteboard, Veranstaltungsunterlagen in elektronischer Form, Nutzung eines e-Learning-Systems, mathematische Anwendungssoftware
Literatur	Proakis, John: Digital Communications, McGraw-Hill Haykin, Simon: Communication Systems, Wiley Haykin, Simon: Digital Communications, Wiley

Modulbezeichnung	Computernetze
Kürzel	Cn
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none"> • 60 h Präsenzstudium (35 h Seminaristischer Unterricht, 25 h Übungen) • 90 h Eigenstudium (30 h Nachbereitung Seminaristischer Unterricht, 20 h Bearbeitung von Übungsaufgaben, 40 h Prüfungsvorbereitung)
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Michael Sammeth
Dozent(in)	Prof. Dr. Michael Sammeth
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	keine
Qualifikationsziele / Kompetenzen	Die Studierenden erhalten Kenntnisse in Struktur und Aufbau von lokalen und Weitverkehrsdatennetzen sowie über die dort eingesetzten Protokolle. Dabei liegt der Schwerpunkt auf Kenntnissen in den Internet-Protokollen, vor allem TCP/IP sowie Routing-Verfahren. Die Teilnehmer erwerben zudem die Fähigkeit zur Analyse von Kommunikationsvorgängen sowie zur Berechnung von Leistungsparametern und Adressierungswerten.
Lehrinhalte	Inhaltlich wird ein Top-Down-Ansatz verfolgt, d.h. die höheren Schichten zuerst behandelt. Die Themen im Einzelnen sind: Grundlagen: Einführung, Netztopologien, ISO/OSI-Referenzmodell, ISO/OSI und TCP/IP, Bandbreite und Performance Anwendungsschicht: Internet-Anwendungen, Protokolle der Anwendungsschicht, Multimedia-Anwendungen

	<p>Transportschicht: Einführung, Struktur des TCP/IP-Modells, Sender/Empfänger-Koordination, Transmission Control Protocol (TCP), Flusskontrolle bei TCP</p> <p>Vermittlungsschicht: Das Internet-Protokoll IP, Einfache IP-Protokolle, IP-Adressierung und Subnetzbildung, Domain Name System und Namensauflösung, IP-Protokoll Version 6, Distanzvektor-Routing, Link State Routing</p> <p>Sicherungsschicht: Rahmenbildung, Fehlererkennung und Fehlerkorrektur, lokale Netze, Ethernet, Medienzugriffsverfahren, WLAN, LAN-Switches (Bridges), Virtuelle Verbindungen, Zellenvermittlung (ATM)</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Min.)
Sonstige Leistungsnachweise	-
Medienformen	<p>Beamer und Tafel/Whiteboard,</p> <p>Elektronische Skripten und Arbeitsunterlagen</p> <p>Selbsttests zur Wiederholung in Moodle</p>
Literatur	<p>L. Peterson, B. Davie: Computernetze. dpunkt.verlag, Heidelberg, 2007</p> <p>J. Kurose, K. Ross: Computernetze. 5. Aufl., Pearson Education, München, 2012</p> <p>A. Tanenbaum: Computer-Netzwerke. 5. Aufl., Pearson Education, München, 2012</p> <p>J. Scherff: Grundkurs Computernetze. Vieweg-Verlag, Wiesbaden, 2010</p>

Modulbezeichnung	Defensive Security Mechanisms
Kürzel	DefSec
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzstudium: 60 h (30 h seminaristischer Unterricht, 30 h Laborprojekte) Eigenstudium: 90 h (45 h Nachbereitung seminaristischer Unterricht & Prüfungsvorbereitung, 45 h Übungsaufgaben & Laborübungen)
Fachsemester	5 oder 7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Henning Maier
Dozent(in)	Prof. Dr. Henning Maier
Sprache	Deutsch, Englisch-Kenntnisse vorteilhaft
Nutzung in anderen Studiengängen	IF, VC, AU, EL
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 (bzw. §6) Abs. 1 SPO des jew. Studiengangs
Inhaltliche Voraussetzungen	Englisch-Kenntnisse vorteilhaft Grundlegende Kenntnisse zur Kryptographie sehr vorteilhaft Grundlegende Kenntnisse zu Computernetzen vorteilhaft Grundlegende Kenntnisse zu Datenbanken vorteilhaft Grundlegende Kenntnisse zu Linux-Systemen vorteilhaft

Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Die Studierenden erlernen grundlegende defensive Sicherheitsmechanismen in IT-Systemen in Theorie und Praxis:</p> <ul style="list-style-type: none"> • Grundlegende defensive Sicherheitsmodelle, -Architekturen und -Mechanismen verstehen und anwenden • Einschätzung, Bewertung und Dokumentation von Schutzmechanismen vornehmen <p>Der praxisbezogene Anteil erfolgt primär in einer virtualisierten Netzwerk-Umgebung auf Linux-basierten Systemen:</p> <ul style="list-style-type: none"> • Einarbeitung in die Nutzung & Konfiguration von Linux-Systemen • Defensive Sicherheitsmechanismen zur Härtung von IT-Systemen praktisch in einer virtualisierten Infrastruktur konfigurieren, testen und dokumentieren • Verwendung der deklarativen Programmiersprache Rego für Policy-as-Code Implementierungen mit dem Open Policy Agent (OPA) <p>Persönlich-soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Studierende präsentieren, visualisieren (Demonstratoren) und diskutieren kritisch untereinander ihre Lösungsansätze (SPA-Praxisaufgaben).
Lehrinhalte	<p>Die betrachteten Konzepte umfassen u.a.:</p> <ul style="list-style-type: none"> • Cybersecurity Frameworks & Terminologies • Access Control: Authentication & Authorization <ul style="list-style-type: none"> ◦ Password Authentication ◦ One-Time-Passwords ◦ Challenge-Response ◦ Biometric User Authentication ◦ Discretionary Access Control (DAC) ◦ Role-Based Access Control (RBAC) ◦ Attribute-Based Access Control (ABAC) ◦ Mandatory Access Control (MAC) • Multi-Level & Multi-Lateral Security <ul style="list-style-type: none"> ◦ Bell-LaPadula Model ◦ Biba Model ◦ Chinese-Wall Model • Network-Security <ul style="list-style-type: none"> ◦ Firewalls, VPN • Trusted Computing <ul style="list-style-type: none"> ◦ Trusted Platform Module (TPM) ◦ Sealing / Binding ◦ Remote Attestation <p>Diese und weitere Konzepte werden für eine Auswahl unterschiedlicher Technologien praxisnah untersucht.</p>
Endnotenbildende Studien- /	Schriftliche Prüfung (Gewicht 50%),

Prüfungsleistungen	Studien- und Projektarbeit (SPA) (Gewicht 50%)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel/Whiteboard, (englisch-sprachige) Vorlesungs- und Arbeitsunterlagen Laborarbeitsplatz mit virtualisierter Netzwerkumgebung Demonstrator-Videoclips (GUI/CLI)
Literatur	<ul style="list-style-type: none"> • W. Stallings, L. Brown, „<i>Computer Security: Principles and Practice</i>“, Pearson, 4th Edition, 2018. • M. Bishop, „<i>Computer Security: Art and Science</i>“, Addison Wesley, 2nd Edition, 2018. • R. Anderson, „<i>Security Engineering</i>“, Wiley, 3rd Edition, 2020. • J. Schwenk, „<i>Sicherheit und Kryptographie im Internet</i>“, Springer Vieweg, 2020. • C. Eckert, „<i>IT-Sicherheit: Konzepte – Verfahren – Protokolle</i>“, DeGruyter, 9. Auflage, 2014. • W. Arthur, et al., „<i>A Practical Guide to TPM 2.0</i>“, Apress, 1st Edition, 2015.

Modulbezeichnung	Digital Hardware Design
Kürzel	DigHaD
Lehrform / SWS	Seminaristischer Unterricht / 2 SWS Praktikum, Übungen / 2 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden, 90 Stunden Eigenarbeit
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Oliver Engel
Dozent(in)	Prof. Oliver Engel
Sprache	Deutsch
Nutzung in anderen Studiengängen	EL, AU
Zulassungsvoraussetzungen	-
Inhaltliche Voraussetzungen	Digitaltechnik, Technische Informatik oder Rechnerarchitekturen
Qualifikationsziele / Kompetenzen	<ol style="list-style-type: none"> 1. Die Studierenden erlangen die Fähigkeit, digitale Schaltungen hinsichtlich Struktur und Verhalten zu modellieren. 2. Die Studierenden beherrschen die Hardwarebeschreibungssprache VHDL und können daraus synthesesfähigen Code erzeugen. 3. Die Studierenden erlernen Methoden, eigene oder fremde digitale Designs zu verifizieren und deren korrekte Arbeitsweise sicherzustellen.
Lehrinhalte	VHDL-Konzepte <ul style="list-style-type: none"> • Strukturelemente: Entity, Architecture, Objekte • Funktionselemente: Prozess, Funktionen und Prozeduren • Modellierung von Speicherelementen sowie kombinatorischen Schaltungen • Datenstrukturen: skalare und zusammengesetzte Datentypen, Arrays, Konstanten, Types und subtypes

	<ul style="list-style-type: none"> • Aufbau von Bibliotheken Modellierung digitaler Hardware: <ul style="list-style-type: none"> • Zustandsautomaten • Speicher: RAM, ROM, Ringspeicher • Tristate-Modellierung, Schnittstellen, Bussysteme • Arithmetikeinheiten • parallele Hardware Verifikation <ul style="list-style-type: none"> • Testbenches Sicherstellung digitaler Beschreibungen Elemente des synchronen Designs
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel
Literatur	<p>Jürgen Reichardt, Bernd Schwarz: VHDL-Synthese, Oldenbourg Verlag, 2015</p> <p>Winfried Gehrke, Marco Winzker: Digitaltechnik: Grundlagen, VHDL, FPGAs, Mikrocontroller, Springer, 2016</p> <p>Pong P.Chu: FPGA Prototyping by VHDL Examples, Wiley, 2008</p>

Modulbezeichnung	Eingebettete Betriebssysteme
Kürzel	EBSy
Lehrform / SWS	4 SWS: – Seminaristischer Unterricht (2 SWS) – Praktikum (2 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 h Präsenz (30 h Seminaristischer Unterricht, 30 h Praktikum) 90 h Eigenarbeit (30 h Nachbereitung des Lehrstoffs, 60 h Vorbereitung und Bearbeitung von Praktikumsaufgaben)
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Markus Hager
Dozent(in)	Prof. Dr. Markus Hager
Sprache	Deutsch
Nutzung in anderen Studiengängen	Bachelor-Studiengänge EL, AU
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Kenntnisse aus den Themenbereichen Rechnerarchitektur, maschinennahe Programmierung (C und Assembler) und Betriebssysteme
Qualifikationsziele / Kompetenzen	Fachkompetenzen: Die Studierenden erlangen fundiertes Fachwissen ... <ul style="list-style-type: none"> – in der Struktur und dem Aufbau von typischen Echtzeitbetriebssystemen für eingebettete Systeme: Sie erkennen, analysieren und bewerten die Komponenten eines eingebetteten Betriebssystems. Sie kennen Beispiele von Betriebssystemen in der Praxis. – in der Ansteuerung der unterliegenden Hardware: Sie beschreiben und erkennen die für die Betriebssystementwicklung relevanten Eigenschaften der Programmiersprache C sowie ARM-Assembler. – in den Mechanismen der Prozessverwaltung eines eingebetteten Betriebssystems: Sie können den Prozesskontext beschreiben, Sie kennen die

	<p>Methoden der Prozessumschaltung (präemptiv, kooperativ).</p> <ul style="list-style-type: none"> – in Echtzeitsystemen: Sie können ein Tasksystem bezüglich der Echtzeitfähigkeit bewerten. Sie kennen typische Schedulingalgorithmen und können diese anwenden. <p>Methodenkompetenzen:</p> <p>Die Studierenden erlangen durch die Durchführung von Projekten im Labor ...</p> <ul style="list-style-type: none"> – die Anwendung von eingebetteten Betriebssystemen: Sie können ein minimales Betriebssystem in der Programmiersprache C anhand von gestuften Aufgabenstellungen eigenständig entwickeln, Fehler finden und korrigieren. – Die Bewertung der SW-Qualität: Sie können nichtfunktionale Eigenschaften, wie z.B. Codelaufzeit, Codegröße und Energieverbrauch analysieren und optimieren. <p>Persönliche Kompetenzen:</p> <p>Die Studierenden erlangen Sozial- und Selbstkompetenz durch ...</p> <ul style="list-style-type: none"> - die teamorientierte Projektarbeit: Sie können im Team Aufgabenstellungen im Umfeld eingebetteter Systeme entwickeln und implementieren. - persönliches Zeitmanagement: Sie können sich im Rahmen der Vor- und Nachbereitung der Seminare, der Praktika und der Prüfung Ihre persönliche Arbeitsweise strukturieren und optimieren.
Lehrinhalte	<ul style="list-style-type: none"> – Einführung und Überblick: Begriffe, Definition – Embedded C: Vertiefung für eingebettete Systeme – Prozessorarchitektur: Hardwaregrundlagen ARM Cortex M-Prozessoren, hardwarenahe Programmierung – Prozesse, Speicher (Text-/Daten-/Stacksegmente) und Prozesskontext – Multitasking und Kontextwechsel – Asynchrone Ereignisse: Interrupts und Timer, präemptives Multitasking – Ansteuerung von I/O-Geräten, einfache Gerätetreiber – Schedulingverfahren, Echtzeit-Schedulingverfahren (RMS, EDF) – Prozesskommunikation und -synchronisation: atomare Operationen, Mutexe, Spinlocks und Semaphore

Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten) und praktische Leistungsnachweise (bearbeitete Aufgaben) im Verhältnis 1:1
Sonstige Leistungsnachweise	–
Medienformen	Beamer / Präsentationsfolien, Praktische Übungen am Rechner / Labor, Moodle
Literatur	<ul style="list-style-type: none"> • Douglas Comer: „Operating System Design: The Xinu Approach“, Second Edition 2015, Chapman and Hall/CRC, ISBN-13: 978-1498712439 • Joseph Yiu: „The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors“, Newnes, 3rd Edition 2013, ISBN-13: 978-0124080829 • Michael Barr, „Programming Embedded Systems in C and C++“, O'Reilly 1999, ISBN: 1-56592-354-5 • H. Wörn, U. Brinkschulte, „Echtzeitsysteme“, Springer, 2005 • D. Zöbel, W. Albrecht, „Echtzeitsysteme: Grundlagen und Technik“ Bonn, Internat. Thomson Publ., 1995 • G. Buzatto: Hard Real-Time Computing Systems. Springer, ISBN 0-387-23137-4. • C.L. Liu, J.W. Layland: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the Association for Computing Machinery (ACM), 20(1), 1973. • M. Homann: OSEK – Betriebssystem-Standard für Automotive und Embedded Systems. MITP-Verlag, 2. Auflage 2005, ISBN 3-8266-1552-2.

Modulbezeichnung	Embedded Softwareentwicklung
Kürzel	EmbSwe
Lehrform / SWS	4 SWS: – Seminaristischer Unterricht (2 SWS) – Übung (1 SWS) und Praktikum (1 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzzeit: 60h, Selbststudium: 90h
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Markus Hager
Dozent(in)	Prof. Dr. Markus Hager
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Programmieren 1 und 2
Qualifikationsziele / Kompetenzen	<p>Fachkompetenzen:</p> <p>Die Studierenden erlangen fundiertes fachliches Grundlagenwissen ...</p> <ul style="list-style-type: none"> - in der Anwendung von Eingebetteten Systemen im industriellen Bereich: Sie erkennen die Abgrenzung zu klassischen Rechnersystemen und die Notwendigkeit von eingebetteten Systemen. - im Aufbau von Eingebetteten Systemen: Sie erkennen die Strukturen von Hard- und Softwarekomponenten moderner Mikrocomputersystemen und Mikrocontrollern und deren Interaktion und können die Eigenschaften beurteilen. <p>Die Studierenden erlernen und üben die Anwendung von Mikrocontrollern, insbesondere ...</p> <ul style="list-style-type: none"> - in der Softwareentwicklung für Mikrocontroller: Sie können gegebene Anforderungen in maschinennahe Programmierung (Assembler) umsetzen.

	<p>Sie können Peripherieeinheiten des Mikrocontrollers in Betrieb nehmen und programmieren.</p> <ul style="list-style-type: none"> - in der Analyse und Umsetzung von Realzeiteigenschaften: Sie können asynchrone Ereignisse (Interrupts) erkennen und programmieren sowie typische Ein- und Ausgabegeräte als Reaktion ansteuern. - die Verwendung moderner Entwicklungs- und Debugging-Werkzeuge: Sie kennen den Softwareentwicklungsprozess und können ein Softwarekonzept im Mikrocontroller umsetzen (Assembler/Compiler, Linker, IDE). <p>Methodenkompetenz:</p> <p>Die Studierenden erlangen durch die Durchführung kleinerer SW-Projekte im Labor ...</p> <ul style="list-style-type: none"> - Strategien zur Problemlösung: Sie können komplexe Sachverhalte einfach darstellen. Sie können Anforderungen analysieren und in eine technische Realisierung umsetzen (Top-Down-Denken). - Methoden der Fehlersuche: Sie können systematisch mit modernen Werkzeugen (Debugger, Oszilloskop, Logikanalysator) das Verhalten von eingebetteten Systemen analysieren und Fehler im Quellcode erkennen und beheben. - Bewertung der Software-Qualität: Sie erkennen die Notwendigkeit strukturierter und dokumentierter Softwareerstellung. Sie können verständlichen Code erstellen und kennen Zusammenhänge zwischen Hochsprachen- und Assemblercode. - die Befähigung zur selbständigen Aneignung und Anwendung (wissenschaftlicher) Erkenntnisse: Sie können einschlägige (englischsprachige) Literatur, insbesondere Datenblätter und Manuals lesen und verstehen. <p>Persönliche Kompetenzen:</p> <p>Die Studierenden erlangen Sozial- und Selbstkompetenz durch ...</p> <ul style="list-style-type: none"> - die teamorientierte Projektarbeit: Sie können im Team Aufgabenstellungen im Umfeld eingebetteter Systeme entwickeln und implementieren. - persönliches Zeitmanagement: Sie können sich im Rahmen der Vor- und Nachbereitung der Seminare, der Praktika und der Prüfung Ihre persönliche Arbeitsweise strukturieren und optimieren.
--	--

Lehrinhalte	<ol style="list-style-type: none"> Einführung: Historie, Begriffserklärung und Definitionen, Abgrenzung und Anforderungen von Eingebetteten Systemen, Anwendung von Mikrocontrollern Rechnerarchitektur: Aufbau und Komponenten eines Mikrocontrollers (ARM-basiert), Hardware-Abstraktion (Programmiermodell), Befehlsverarbeitung, Funktionsweise des Rechnerkerns und der Peripherie, Speicherorganisation, Stack, Registersatz, Interruptverarbeitung Einführung in die Assemblerprogrammierung: Aufbau Assembleranweisung, Befehlssatz, Befehlsgruppen (Arithmetische Befehle, Logikbefehle, Sprungbefehle, ...), Adressierungsarten, Rechnerarithmetik und Zahlensysteme, Elementare Programmstrukturen (Schleifen, Unterprogramme, Verzweigungen), Echtzeitverhalten (synchrones und asynchrones Software-Design/Interrupts), Hochsprachenbezug (C, Compiler) Peripherie eines Mikrocontrollers: Digitale Ein-/Ausgabe, Interrupts, Timer und Zähler, serielle Schnittstellen (UART, Bussysteme, Zugriffsverfahren), analoge Signale und Wandlung, externe Speicherschnittstellen Beispielprojekte / Praktischer Einsatz (Labor): Verwendung moderner Entwicklungswerkzeuge (IDE, Debugger); Konfiguration eines aktuellen praxisorientierten Systems aus vorgefertigten Hardwarekomponenten; Anwendung der hardwarenahen (Assembler-) Programmierung für eine komplexe Anwendung unter Einsatz verschiedener Hardwarekomponenten (jährlich wechselnd): (z.B. Keyboards, LCD-Displays, GPS- Empfänger, RFID-Devices, Bluetooth-Transmitter, Messwandler, Schrittmotor-Ansteuerung, DCF-Empfänger, Druckwerk-Ansteuerung)
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung 90 Min. und prStA (Projekte und Testate)
Sonstige Leistungsnachweise	keine
Medienformen	Beamer / Präsentationsfolien, theoretische Übungsblätter, Praktische Übungen am Rechner / Labor
Literatur	<p>Helmut Bähring, „Anwendungsorientierte Mikroprozessoren – Mikrocontroller und Signalprozessoren“, 3. Auflage, Springer, 2010</p> <p>T. Flik, H. Liebig, „Mikroprozessortechnik und Rechnerstrukturen“, Springer, 2005</p> <p>U. Brinkschulte, T. Ungerer „Mikrocontroller und Mikroprozessoren“, Springer, 2007</p>

	<p>K. Wüst, „Mikroprozessortechnik“, Vieweg+Teubner, 2011</p> <p>W. Stallings, „Computer Organization and Architecture“, Pearson, 2016</p> <p>Joseph Yiu, „The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors“, Newnes, 3rd Edition 2013, ISBN-13: 978-0124080829</p> <p>Jonathan M. Valvano, „Embedded Systems: Introduction to ARM Cortex-M Microcontrollers“, CreateSpace Independent Publishing, 2nd Ed. 2012, ISBN-13: 978-1477508992</p>
--	--

Modulbezeichnung	E-Entrepreneurship
Kürzel	EEnt
Lehrform / SWS	Seminaristischer Unterricht inkl. Übungen / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden, 90 Stunden Eigenarbeit
Fachsemester	6 oder 7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Thomas Wieland
Dozent(in)	Herr Johannes Schmölz
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 SPO
Inhaltliche Voraussetzungen	Gute Kenntnisse in Webtechnologien bzw. mobilen Technologien.
Qualifikationsziele / Kompetenzen	Studierende sollen die Grundlagen der Unternehmensgründung auf Basis digitaler Geschäftsprozesse kennen und verstehen. Studierende sollen in der Lage sein, einen einfachen Prototypen und das dazugehörige Geschäftsmodell zu entwickeln.
Lehrinhalte	Die Lehrveranstaltung verbindet aktuelle Konzepte wie Lean Startup, Design Thinking, Business Model Canvas und Agile Development zu einem umfassenden Vorgehensmodell, mit dem aus Ideen und Innovationen erfolgreiche Unternehmen werden.
Endnotenbildende Studien- / Prüfungsleistungen	Studienarbeit
Sonstige Leistungsnachweise	
Medienformen	Beamer, Tafel
Literatur	Blank, S. (2014): Das Handbuch für Startups, 1. Aufl., O'Reilly

Modulbezeichnung	Fortgeschrittene Webtechnologien
Kürzel	FWebt
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon 60 h Präsenz (30 h seminaristischer Unterricht, 30 h Praktikum) und 90 h Eigenarbeit (30 h Vor- und Nachbereitung des Lehrstoffs, 40 h Bearbeitung von Übungs- und Projektaufgaben, 20 h Prüfungsvorbereitung)
Fachsemester	6 oder 7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Prof. Dr. Jürgen Terpin
Sprache	Deutsch
Nutzung in anderen Studiengängen	BA Visual Computing
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Grundkenntnisse in HTML, CSS und JavaScript sowie grundlegende Kenntnisse des Document Object Model (DOM), der asynchronen Client-Server-Kommunikation, von HTTP und JSON. Grundlegende Programmierkenntnisse sowie ein sicherer Umgang mit einer Entwicklungsumgebung und einer Versionsverwaltung.
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • erläutern die Architektur moderner Webanwendungen und grenzen die Verantwortlichkeiten von Browser, Frontend, Webserver, Backend, Programmierschnittstelle und Datenbank voneinander ab, • analysieren unterschiedliche Ansätze zur Erzeugung und Auslieferung von Weboberflächen und beurteilen deren Eignung für konkrete Anwendungsszenarien,

	<ul style="list-style-type: none"> • konzipieren und implementieren komponentenbasierte, interaktive Weboberflächen mit React, • strukturieren größere Frontend-Anwendungen mithilfe von Komponenten, Routing, Zustandsverwaltung, Formularverarbeitung und wiederverwendbaren Diensten, • entwerfen und implementieren serverseitige Programmierschnittstellen für Webanwendungen und wenden dabei geeignete HTTP-Methoden, Statuscodes und Datenformate an, • realisieren die asynchrone Kommunikation zwischen Frontend und Backend einschließlich Validierung, Fehlerbehandlung und Rückmeldung an die Nutzenden, • modellieren Anwendungsdaten, binden eine Datenbank an ein Backend an und implementieren grundlegende persistente Operationen, • setzen Werkzeuge für Paketverwaltung, Build-Prozesse, Versionsverwaltung, automatisierte Tests, Fehleranalyse und Qualitätssicherung sachgerecht ein, • berücksichtigen bei der Entwicklung nicht-funktionale Anforderungen, insbesondere Sicherheit, Barrierefreiheit, Bedienbarkeit und Performanz, • vergleichen technische Lösungsalternativen und begründen Architektur- und Technologieentscheidungen nachvollziehbar, • entwickeln eine zusammenhängende Full-Stack-Webanwendung von der Anforderungsanalyse bis zu einer lauffähigen und dokumentierten Lösung. <p>Persönliche und soziale Kompetenzen:</p> <p>Innerhalb des Praktikums und des seminaristischen Unterrichts bearbeiten die Studierenden Aufgaben und ein zusammenhängendes Entwicklungsprojekt in Kleingruppen von in der Regel zwei bis drei Personen.</p> <p>Dabei erwerben und vertiefen sie insbesondere die Fähigkeit,</p> <ul style="list-style-type: none"> • Arbeitsabläufe, Rollen, Ressourcen und Termine innerhalb eines Entwicklungsteams zu planen und zu koordinieren,
--	---

	<ul style="list-style-type: none"> • eine komplexe Aufgabenstellung in geeignete Teilaufgaben zu zerlegen und Verantwortlichkeiten nachvollziehbar zuzuordnen, • technische Entscheidungen innerhalb der Gruppe zu diskutieren, zu dokumentieren und gegenüber Dritten zu begründen, • Quellcode und Arbeitsergebnisse anderer Gruppenmitglieder konstruktiv zu prüfen und Rückmeldungen adressatengerecht zu formulieren, • Konflikte und unterschiedliche fachliche Einschätzungen möglichst eigenständig und lösungsorientiert zu bearbeiten, • den eigenen Lern- und Entwicklungsprozess zu reflektieren und sich selbstständig in neue Werkzeuge und Bibliotheken einzuarbeiten, • Ergebnisse strukturiert zu präsentieren und in einem Fachgespräch zu verteidigen, • KI-gestützte Entwicklungswerkzeuge transparent, kritisch und verantwortungsvoll einzusetzen sowie erzeugte Ergebnisse fachlich zu prüfen.
Lehrinhalte	<ul style="list-style-type: none"> • Architektur moderner Webanwendungen sowie clientseitiges und serverseitiges Rendering • moderne JavaScript- und TypeScript-Entwicklung, Module und asynchrone Programmierung • komponentenbasierte Frontend-Entwicklung mit React • Routing, Zustandsverwaltung, Formulare und Validierung • REST-orientierte Programmierschnittstellen mit HTTP und JSON • Backend-Entwicklung mit Node.js und Express • Datenmodellierung und Persistenz mit SQLite • Tests, Sicherheit, Barrierefreiheit und Performanz • Build-Prozess mit Vite, Versionsverwaltung mit Git und Zusammenarbeit über GitLab • durchgängiges Full-Stack-Projekt von der Anforderungsanalyse bis zur Präsentation
Endnotenbildende Studien- / Prüfungsleistungen	Praktische Studienarbeit in Kleingruppen
Sonstige Leistungsnachweise	keine
Medienformen	Beamer beziehungsweise Präsentationsfolien, Tafel, digitale Lehrmaterialien, E-Learning-Medien und Quellcodebeispiele.

Literatur	<ul style="list-style-type: none"> • Ackermann, Philip (2023): Fullstack-Entwicklung. Das Handbuch für Webentwickler, 2., aktualisierte und erweiterte Auflage, Rheinwerk Computing, Bonn. • Ackermann, Philip (2025): JavaScript. Das umfassende Handbuch, 4. Auflage, Rheinwerk Computing, Bonn. • Springer, Sebastian (2025): Node.js. Das umfassende Handbuch, 5., aktualisierte und erweiterte Auflage, Rheinwerk Computing, Bonn. • Springer, Sebastian (2026): React. Das umfassende Handbuch, 3., aktualisierte und erweiterte Auflage, Rheinwerk Computing, Bonn.
-----------	--

Modulbezeichnung	Grundlagen der Wirtschaftsinformatik
Kürzel	GWi
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none"> • 60 h Präsenz (30 h Seminaristischer Unterricht, 30 h Praktikum) • 90 h Eigenarbeit (30 h Vor- und Nachbereitung des Lehrstoffs, 30 h Lösung von Übungsaufgaben, 30 h Prüfungsvorbereitung)
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Prof. Dr. Jürgen Terpin
Sprache	Deutsch
Zuordnung zum Curriculum	Wahlpflichtmodul
Nutzung in anderen Studiengängen	Informatik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 SPO
Inhaltliche Voraussetzungen	Grundkonzepte der Informatik und der Betriebswirtschaft, etwa aus den Veranstaltungen <ul style="list-style-type: none"> • Grundlagen der Informatik • Betriebswirtschaftslehre
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: <ul style="list-style-type: none"> • Die Studierenden können den Gegenstand der Wirtschaftsinformatik (WI) und ihren interdisziplinären Ansatz erläutern. Sie sind in der Lage die grundlegenden Teilbereiche der Wirtschaftsinformatik zu klassifizieren und können deren wesentliche Inhalte reproduzieren. • Sie können die grundsätzlichen Bestandteile, Aufgaben und Arten von Informations- bzw. Anwendungssystemen skizzieren und können Systeme aus der Praxis entsprechend einordnen.

	<ul style="list-style-type: none"> • Sie können die Herausforderungen bei Planung, Entwicklung/Beschaffung, Implementierung und Betrieb von Informations- bzw. Anwendungssystemen erklären und sind in der Lage, entsprechende Beispiele/Aufgaben aus der Praxis zu bearbeiten. • Die Studierenden können die Bedeutung funktionsübergreifender Unterstützung von Geschäftsprozessen durch integrierte Standardsoftware erläutern. • Die Studierenden können die grundsätzliche Funktionsweise der in Unternehmen eingesetzten ERP-Systeme erklären. • Sie können darstellen und ableiten, wo WI-Wissen in der Praxis benötigt und eingesetzt wird (charakteristische Arbeitsfelder). <p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • Innerhalb des Praktikums werden Arbeitsaufgaben in Kleingruppen (3-4 Personen) bearbeitet. Die gemeinsame Bearbeitung erfolgt dabei teilweise auch außerhalb der Hochschule (in Präsenz oder online). • Die Studierenden erwerben bzw. verbessern personale Kompetenzen indem sie beispielsweise innerhalb des Gruppenarbeitskontextes gemeinsam <ul style="list-style-type: none"> ◦ die zur Zielerreichung notwendigen Abläufe, Ressourcen etc. organisieren/koordinieren, ◦ die Gesamtaufgabe in Teilaufgaben zerlegen und diese auf die Gruppenmitglieder verteilen, ◦ sich gegenseitig informieren, abstimmen etc. und entsprechende (digitale) Werkzeuge hierfür einsetzen, ◦ Konflikte möglichst eigenständig auflösen, ◦ komplexe Sachverhalte in der Gruppe diskutieren und strukturieren sowie zielgerichtet und adressatenbezogen im Plenum darstellen/präsentieren, ◦ Beim Einsatz von ERP-Software unterschiedliche Benutzerrollen einnehmen.
Lehrinhalte	<ul style="list-style-type: none"> • Gegenstand, fachliche Einordnung und Methoden der WI • Informations-/Anwendungssysteme • Wichtige Technologien/technologische Trends im Bereich IT • Entwicklungsschritte der IT(-Organisation) inkl. Digitalisierung

	<ul style="list-style-type: none"> • Betriebliche Funktionsbereiche und Geschäftsprozesse und deren Unterstützung durch Anwendungssysteme • Integration von Informations-/Anwendungssystemen • Funktionsübergreifend integrierte Standardsoftware in der Ausprägung "ERP-Systeme" (Enterprise Resource Planning)
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Overhead, Tafel, E-Learning Medien
Literatur	<ul style="list-style-type: none"> • Abts, D.; Müller, W. (2013): Grundkurs Wirtschaftsinformatik – Eine kompakte und praxisorientierte Einführung, 8. Aufl., Wiesbaden. • Hansen, H. R.; Mendling, J.; Neumann, G. (2009): Wirtschaftsinformatik – Grundlagen und Anwendungen, 11. Aufl., Berlin u.a. • Hesseler, M.; Görtz, M. (2008): Basiswissen ERP-Systeme – Auswahl, Einführung & Einsatz betriebswirtschaftlicher Standardsoftware, 1. korrigierter Nachdruck, Herdecke. • Kurbel, K. (2016): Enterprise Resource Planning und Supply Chain Management in der Industrie. Von MRP bis Industrie 4.0. 8. Aufl. Berlin/Boston. • Mertens, P.; Bodendorf, F.; König, W.; Picot, A.; Schumann, M.; Hess, T. (2012): Grundzüge der Wirtschaftsinformatik, 11. Aufl., Berlin.

Modulbezeichnung	IT-Sicherheit
Kürzel	ITS
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzstudium: 60 h (30 h Seminaristischer Unterricht, 30 h Laborübungen) Eigenstudium: 90 h (60 h Nachbereitung seminaristischer Unterricht / Prüfungsvorbereitung, 30 h Laborübungen)
Fachsemester	4 oder 6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Henning Maier
Dozent(in)	Prof. Dr. Henning Maier
Sprache	Deutsch
Nutzung in anderen Studiengängen	Elektrotechnik AU, EL, EN, VC
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Kenntnisse aus „Computernetze“ Kenntnisse aus „Diskrete Mathematik“
Qualifikationsziele / Kompetenzen	Die Studierenden erhalten Kenntnisse über die Grundbegriffe der Kryptographie und der Herausforderungen und Maßnahmen der IT-Sicherheit. Insbesondere sollen sie die mathematischen Hintergründe aktueller kryptographischer Verfahren kennen und verstehen. Sie sollen die Funktionsweise dieser Verfahren sowie von Hashfunktionen, Signatur- und Authentisierungsverfahren verstehen und sie auch anwenden können. Zudem sollen Studierende Grundlagen der offensiven IT-Sicherheit verstehen und in vorbereiteten Szenarien anwenden.
Lehrinhalte	Klassische Verschlüsselungsverfahren Grundbegriffe der Kryptographie Grundlagen Kryptoanalyse

	<p>Endliche Zahlenmengen und Restklassen</p> <p>Blockchiffren und Operationsmodi</p> <p>Rechnung mit Potenzen modulo n</p> <p>Public-Key-Kryptographie</p> <p>kryptographische Hashfunktionen</p> <p>Digitale Signaturen</p> <p>Anwendungsprotokolle</p> <p>Anwendungen und offensive IT-Sicherheit</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten), praktische Studienarbeit (Gewicht 1:1)
Sonstige Leistungsnachweise	-
Medienformen	<p>Beamer und Tafel/Whiteboard</p> <p>Elektronische Skripten und Arbeitsunterlagen</p> <p>Jupyter-Notebooks</p> <p>Praktikum mit virtualisierten Übungsumgebungen (z.B. VirtualBox, Docker/Podman)</p>
Literatur	<ul style="list-style-type: none"> ● C. Paar, J. Pelzl: Kryptografie verständlich, Springer Vieweg, 2016 ● J. Ziegenbalg: Elementare Zahlentheorie, Springer Spektrum, 2. Auflage, 2015 ● S. Spitz, M. Pramateftakis, J. Swoboda: Kryptographie und IT-Sicherheit, 2. Auflage, Vieweg Teubner, 2011 ● J. Buchmann: Einführung in die Kryptographie. Springer Verlag, 6. Auflage, 2016 ● C. Eckert: IT-Sicherheit. De Gruyter Oldenbourg, 11. Auflage, 2023 ● A. Menezes, P. Oorschot und S. Vanstone. CRC Press, Handbook of Applied Cryptography, 5. Aufl. 2001. ● Computer Security: Principles and Practice, W. Stallings, L. Brown, 4. Edition, 2018.

Modulbezeichnung	Robotik
Kürzel	Ro
Lehrform / SWS	Vorlesung (2 SWS) + Projekt (2 SWS)
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h 60 h Präsenz, 90 h Eigenarbeit (60 h Vor- und Nachbereitung des Lehrstoffs, 30 h Prüfungsvorbereitung)
Fachsemester	6 oder 7
Angebotsturnus	Jährlich
Dauer des Moduls	Einsemestrig
Modulverantwortliche(r)	Prof. Dr. habil. Kolja Kühnlenz
Dozent(in)	Prof. Dr. habil. Kolja Kühnlenz
Sprache	Deutsch
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Analysis, Diskrete Mathematik, Programmieren 1/2, Fortgeschrittene Programmierung
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <p>Nach der Veranstaltung kennen und verstehen die Studierenden die grundlegenden Methoden zur Modellierung, Analyse und Steuerung von Robotern. Sie können die Methoden auf verschiedene Systeme der manipulierenden oder mobilen Robotik anwenden. Die Studierenden kennen und verstehen die Funktionsprinzipien verschiedener Sensoren in der Robotik.</p> <p>Persönliche und soziale Kompetenzen:</p> <p>Die Studierenden erwerben bzw. verbessern personale Kompetenzen indem sie beispielsweise innerhalb des Gruppenarbeitskontextes gemeinsam</p> <ul style="list-style-type: none"> • die zur Zielerreichung notwendigen Abläufe, Ressourcen etc. organisieren/koordinieren, • die Gesamtaufgabe in Teilaufgaben zerlegen und diese auf die Gruppenmitglieder verteilen, • sich gegenseitig informieren, abstimmen etc. und entsprechende (digitale) Werkzeuge hierfür einsetzen,

	<ul style="list-style-type: none"> • Konflikte möglichst eigenständig auflösen, • komplexe Sachverhalte in der Gruppe diskutieren und strukturieren sowie zielgerichtet und adressatenbezogen im Plenum darstellen/präsentieren.
Lehrinhalte	<p>Das Modul besteht aus einem Vorlesungsteil und einem Projektteil. Im Vorlesungsteil werden folgende Inhalte behandelt:</p> <ul style="list-style-type: none"> • Roboterarme und -fahrzeuge • Räumliche Objektrepräsentation und Transformationen • Kinematik-Modelle von Manipulatoren und Roboterfahrzeugen (direkte und inverse Kinematik, differentielle Kinematik, Jacobi-Matrix, Redundanz und Singularitäten, Prinzip der virtuellen Arbeit) • Kinematische Bahn- und Pfadplanung • Steuerungsarchitekturen • bildgebende Sensoren • Rechen- und Entwurfsübungen <p>Im Projektteil bearbeiten die Studierenden in kleinen Gruppen jeweils ein Projekt an einem mobilen Roboter, um eine ausgewählte sensorgeführte Aufgabenbewältigung, z.B. im Bereich der Indoor- oder Outdoornavigation, exemplarisch studienbegleitend umzusetzen.</p>
Endnotenbildende Studien- / Prüfungsleistungen	Klausur (90 Minuten) und Projektarbeit
Sonstige Leistungsnachweise	-
Medienformen	Präsentation mit Beamer, Gruppenarbeit, E-Learning Medien
Literatur	J.J. Craig, Introduction to Robotics: Mechanics and Control, Prentice Hall.

Modulbezeichnung	Secure Software Engineering
Kürzel	SecSE
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzstudium: 60 h (30 h Seminaristischer Unterricht, 30 h Laborprojekte) Eigenstudium: 90 h (60 h Nachbereitung seminaristischer Unterricht, 30 h Laborübungen)
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Henning Maier
Dozent(in)	Prof. Dr. Henning Maier
Sprache	Deutsch, Englisch-Kenntnisse vorteilhaft
Nutzung in anderen Studiengängen	Elektrotechnik AU, EL, EN, Visual Computing VC
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 SPO
Inhaltliche Voraussetzungen	Kenntnisse aus „Computernetze“ Kenntnisse aus „IT-Sicherheit“ Kenntnisse aus „Informatik“ Kenntnisse aus „Programmierung“ Kenntnisse aus „Englisch“
Qualifikationsziele / Kompetenzen	fachlich methodische Kompetenzen: 1. Analyse-Methoden und Terminologie aus der IT-Sicherheit verstehen und anwenden 2. Verwendung von Design-Techniken, - Strategien und Software-Werkzeugen zur Erreichung von Sicherheitszielen und zur Steigerung der Code-Robustheit 3. Umgang mit technischen Standards & Dokumentation Persönlich-soziale Kompetenzen: Zusammenarbeit in agilen Entwicklungsteams inkl. Test,

	Dokumentation und gegenseitigem Code-Review.
Lehrinhalte	<p>Sicherheitskonzepte & Frameworks</p> <p>Sichere Architekturen & Entwicklung</p> <p>Sichere Programmierung mit Rust</p> <p>Test von Sicherheitsfunktionalität</p>
Endnotenbildende Studien- / Prüfungsleistungen	Studien- und Projektarbeit, mit zusätzlich getätigten Übungsaufgaben können 15 % Bonus erzielt werden
Sonstige Leistungsnachweise	-
Medienformen	<p>Beamer und Tafel/Whiteboard,</p> <p>Elektronische Skripten und Arbeitsunterlagen</p> <p>PC-Laborprojekt mit virtualisierter Übungsumgebung</p>
Literatur	<ul style="list-style-type: none"> ● C. Eckert: IT-Sicherheit. Oldenbourg-Verlag, 2009 ● „Common Criteria for Information Security Evaluation – Part 1: Introduction and general model, CC:2022 Release 1“, Nov. 2022. ● „Common Criteria for Information Security Evaluation – Part 2: Security functional components, CC:2022 Release 1“, Nov. 2022. ● „Threat Modeling: Designing for Security“, A. Shostack, 2014. ● Computer Security: Principles and Practice, W. Stallings, L. Brown, fourth Ed., 2018. ● https://www.rust-lang.org/learn

Modulbezeichnung	Shell und Prozesse
Kürzel	ShuP
Lehrform / SWS	Seminaristischer Unterricht / 2 SWS; Praktikum / 2 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden; 90 Stunden Eigenarbeit
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Dieter Wißmann
Dozent(in)	Prof. Dr. Dieter Wißmann
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung in den zweiten Studienabschnitt gemäß SPO B IF vom 29.07.2020, §5 Abs. 1.
Inhaltliche Voraussetzungen	Kenntnis der Konzepte von prozeduralen Programmiersprachen, vorzugsweise der Programmiersprache C
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden sollen in der Praxis mit Kommandos in der Kommandozeile/Shell umgehen können. • Sie sollen den Zusammenhang von Kommandos und Prozessen, den Zusammenhang von Prozessen und Kindprozessen sowie die Verkettung von Prozessen erklären können. • Sie sollen Prozesse programmiertechnisch anwenden können, z. B. neue Prozesse in eigenen Programmen erzeugen und diese steuern. • Sie sollen in eigenen Programmen korrekt und sicher das Thema Kommunikation zwischen Prozessen behandeln zu können. • Sie sollen wiederkehrende Aufgaben automatisieren können, insbesondere in der Administration. • Sie sollen entscheiden können, für welche Aufgaben die grafische Bedienoberfläche eines Computersystems und für welche die textuelle Bedienoberfläche in der Kommandozeile vorzuziehen

	ist.
Lehrinhalte	<ul style="list-style-type: none"> • Umgang mit der Kommandozeile/Shell in Linux • Systemkommandos für Dateien und das Dateisystem, sowie für Prozesse. • Kommandos für das Anzeigen von, Suchen in und Bearbeiten von Textdateien. • Ein-/Ausgabekanäle und Verkettung von Prozessen durch Umlenkung von Kanälen/Pipes • Shell-Programmierung mit der Bash <ul style="list-style-type: none"> o Automatisierung von Kommandofolgen o Variablen o Fallunterscheidungen, Schleifen o Verarbeitung von Zeichenketten • Prozesse in Linux <ul style="list-style-type: none"> o Erzeugung und Beendigung o Wertübergabe und Wertrückgabe o Signale zum Steuern von Prozessen o Vordergrund- und Hintergrundprozesse • Interprozesskommunikation in Linux <ul style="list-style-type: none"> o Umlenkung von Kanälen/Pipes (in der Shell) o Semaphoren (im C-Programm, System-API) o Shared Memory (im C-Programm, System-API) • Entwicklungswerkzeuge <ul style="list-style-type: none"> o Generierung ausführbarer Code (C-Compiler) o Fehlersuche
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min) und praktische Leistungsnachweise im Rahmen einer praktische Studienarbeit im Verhältnis 1:1
Sonstige Leistungsnachweise	-
Medienformen	<p>Beamer, Tafel;</p> <p>Lernmanagementsysteme (Moodle), Elektronisches Skript und elektronische Arbeitsunterlagen;</p> <p>PC-Systeme;</p>
Literatur	<p>Antonova R, Slaveva V., Slavova T.; Grundlagen und Praxis der Bash- und C-Programmierung in Unix/Linux, Pearson Deutschland GmbH, München, 2022.</p> <p>Cooper M: Advanced Bash-Scripting Guide; 2014; http://tldp.org/LDP/abs/html/index.html (abgerufen 26.10.2022)</p> <p>Plöttner J., Wendzel S.: Linux, Der Grundkurs; Rheinwerk, Bonn, 2021.</p> <p>Wolf J., Kania S.: Linux-UNIX-Programmierung; 4. Auflage,</p>

	<p>Rheinwerk, Bonn, 2016.</p> <p>Wolf J., Kania S., Sommer F.: Shell-Programmierung; 7. Auflage, Rheinwerk, Bonn, 2022.</p>
--	---

Modulbezeichnung	Software-Anforderungen und -Modellierung
Kürzel	SAM
Lehrform / SWS	Seminaristischer Unterricht, Übung / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h davon 60 h Präsenz (Seminaristischer Unterricht: 45 h, Praktikum: 15 h) 90 h Eigenarbeit (Vor- und Nachbereitung des Lehrstoffs: 20h, Projektarbeiten: 40h, Prüfungsvorbereitung: 30h)
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Landes
Dozent(in)	Prof. Dr. Landes, Dr. Sedelmaier
Sprache	Deutsch
Nutzung in anderen Studiengängen	Bachelor Industriewirtschaft
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen Software Engineering
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die Bedeutung und charakteristischen Merkmale von Anforderungen verstehen, • die Komplexität und Zusammenhänge von anforderungsrelevanten Aspekten verstehen, • verbreitete Ansätze und Methoden zur Erhebung, Spezifikation, Dokumentation, Priorisierung, Validierung / Qualitätssicherung und Verfolgbarkeit funktionaler und nicht-funktionaler Anforderungen verstehen und anwenden können, • die Rolle von Kommunikation im Requirements Engineering verstehen und entsprechend handeln können • die Rolle von Geschäftsprozessen als Anforderungsquelle verstehen,

	<ul style="list-style-type: none"> • verbreitete Ansätze zur Aufwands- und Kostenschätzung verstehen und anwenden können.
Lehrinhalte	<ul style="list-style-type: none"> • Anforderungserhebung (Requirements Elicitation) <ul style="list-style-type: none"> ◦ Stakeholderanalyse und -management ◦ Kommunikationstechniken ◦ Techniken zur Anforderungserhebung • Spezifikation und Priorisierung von Anforderungen <ul style="list-style-type: none"> ◦ Prozessmodellierung ◦ Spezifikation funktionaler und nicht-funktionaler Anforderungen ◦ Priorisierungstechniken • Verfolgbarkeit von Anforderungen (Requirements Traceability) • Aufwands- und Kostenschätzung <ul style="list-style-type: none"> ◦ Schätzung funktionaler Größe ◦ Algorithmische Schätzverfahren
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Tafel, Beamer, Visualizer, Video, Modellierungswerkzeuge, Metaplankarten, Pinnwände
Literatur	<p>Rupp, C.: Requirements-Engineering und -Management. Hanser, 6. Auflage, 2015</p> <p>Wieggers K., Beatty, J.: Software Requirements. Microsoft Press, Redmond, 3. Auflage, 2013</p> <p>Drescher, A., Koschmider, A., Oberweis, A.: Modellierung und Analyse von Geschäftsprozessen. De Gruyter Oldenbourg, 2017</p> <p>Gadatsch, A.: Grundkurs Geschäftsprozess-Management. Vieweg-Teubner, 7. Auflage, 2012</p> <p>McConnell, S.: Software Estimation. Microsoft Press, Redmond, 2006</p> <p>Weiterführende Spezialliteratur für die verschiedenen Kapitel</p>

Modulbezeichnung	Software-Architekturen und -Testen
Kürzel	SAT
Lehrform / SWS	Seminaristischer Unterricht, Übung / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h davon 60 h Präsenz (Seminaristischer Unterricht: 45 h, Praktikum: 15 h) 90 h Eigenarbeit (Vor- und Nachbereitung des Lehrstoffs: 20h, Projektarbeiten: 40h, Prüfungsvorbereitung: 30h)
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Pfeiffer
Dozent(in)	Prof. Pfeiffer
Sprache	Deutsch
Zuordnung zum Curriculum	Fachwissenschaftliches Wahlpflichtfach
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung in den zweiten Studienabschnitt gemäß SPO B IF vom 29.02.2020, §5
Inhaltliche Voraussetzungen	Grundlagen Software Engineering
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die Entwurfsprinzipien für Software-Architekturen kennen, verstehen und anwenden können, • Design-Patterns kennen, verstehen und auf ausgewählte Problemstellungen anwenden können, • die Dokumentationsarten von Architekturen kennen, • Test-Grundlagen kennen, verstehen und anwenden können, • systematische Testtechniken zur Herleitung und Entwicklung von Tests und Testfälle für verschiedene Artefakte (z.B. Anforderungen, Code, Modelle) kennen, verstehen und anwenden können.

	<p>Persönliche-Soziale Kompetenzen:</p> <p>Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • einzelne Entwurfs- und Testaufgaben in Kleingruppen teamorientiert zu bearbeiten, • verschiedene Lösungen und Lösungsvarianten im Plenum vorzustellen, zu diskutieren und kritisch zu reflektieren, • Zeitressourcen einzuteilen und mit aufgabenbezogenen Stressoren umzugehen und ihre Ausdauer zu trainieren, • sich innerhalb einer Klein-Gruppe abzustimmen und gemeinsame technische Lösungen mit anderen Teams auszuhandeln und zu erreichen.
Lehrinhalte	<ul style="list-style-type: none"> • Grundlagen Software-Architekturen • Grundlegende Software-Entwurfskonzepte • Dokumentation von Software Architekturen • Design Patterns • Ausgewählte Design Aspekte • Grundlagen Software Testen • Test Generierung für Requirements und Code • Test Generierung für Code • Testen von objektorientierten Systemen
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overheadprojektor, E-Learning Medien
Literatur	<p>Starke, Gernot; Effektive Software Architekturen Hanser Verlag jeweils in der neusten Auflage</p> <p>Fowler, M.: Patterns of enterprise application architecture Addison Wesley 2003</p> <p>Gamma, E., Helm R., Johnson R., Vlissides J.: Entwurfsmuster Addison Wesley 1995.</p> <p>Spillner, A; Linz, T.; Basiswissen Softwaretest dpunkt.verlag jeweils in der neusten Auflage</p> <p>Diverse Spezialliteratur für die verschiedenen Kapitel</p>

Modulbezeichnung	Softwareentwurf in der Automatisierungstechnik
Kürzel	SwAu
Lehrform / SWS	Seminaristischer Unterricht (2 SWS), Übung (1 SWS), Praktikum (1 SWS) / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzzeit: 60h, Selbststudium: 90h
Fachsemester	7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr.-Ing. Matthäus Brela
Dozent(in)	Prof. Dr.-Ing. Matthäus Brela
Sprache	deutsch
Nutzung in anderen Studiengängen	AU, EN, EL, IF, VC
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5
Inhaltliche Voraussetzungen	Grundlagen der Digitaltechnik, Automatentheorie, Zustandsgraphen, Steuerungs- und Regelungstechnik, Kenntnis einer höheren Programmiersprache
Qualifikationsziele / Kompetenzen	<ul style="list-style-type: none"> • Die Studierenden sollen die Methoden und Programmier Techniken der industriellen Steuerungstechnik kennenlernen und einfache Automatisierungsaufgaben in den verschiedenen Programmiersprachen der IEC 61131 selbständig lösen können. • Studierende sind in der Lage ein Steuerungsprogramm zu strukturieren, Modellierungssprachen der UML anzuwenden, objektorientiert zu programmieren, wiederverwendbaren Code zu schreiben, Bibliotheken zu erstellen und Programmierungsfehler zu bereinigen. • Kennenlernen der Funktionsweise serieller Datenkommunikation in der Automatisierungstechnik und Projektierung einer Buskommunikation. • Kennenlernen der Mensch-Maschine-Schnittstelle und der Methoden für Projektierung und Erstellung von Bedienoberflächen für Industriesteuerungen
Lehrinhalte	<ul style="list-style-type: none"> • Steuerungstechnik

	<p>Konfiguration von Steuerungen, Kommunikation in der Automatisierungstechnik, Feldbussysteme, verteilte Systeme nach IEC 61499, Methoden des Programmierens nach IEC61131-3 in Strukturiertem Text.</p> <ul style="list-style-type: none"> • Entwurf von Visualisierungen <p>Bedienen und Beobachten, Visualisierungselemente, Elementverknüpfung, Steuerung mit Visualisierungen, Überwachung, Analyse, erstellen einfacher Bedienoberflächen.</p> <ul style="list-style-type: none"> • Softwareentwurf <p>V-Modell, Verwendung der Ablaufsprache zur Schrittkettenprogrammierung, Erstellung von Klassendiagramm und deren Anwendung, Erstellung von Zustandsdiagramm und deren Anwendung.</p> <ul style="list-style-type: none"> • Objektorientierte Programmierung <p>Aufbau, Entwurf und Programmierung von Klassen, Anwendung von Objekte, Kapselung, Vererbung, Ableitung, Zugriffsmodifizierungen, Konstruktoren, Destruktoren, Properties, Referenzen, Interfaces, virtuelle und abstrakte Methoden, Rezeptverwaltung.</p> <ul style="list-style-type: none"> • Wiederverwendbarkeit <p>Erstellung von Bibliotheken, Fehlerhandhabung, Verwendung von Pragmas, Programmierrichtlinien</p> <ul style="list-style-type: none"> • Praktikum: <p>Kommunikation (Protokolle), Greifersteuerung (Visualisierung), Greifersteuerung (Ablaufsteuerung), Betriebszustände (Zustandsdiagramm), Objektorientierte Programmierung (Klassendiagramm), Bibliotheken (Interface), Inbetriebnahme.</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung 90 Min. und praktische Leistungsnachweis
Sonstige Leistungsnachweise	keine
Medienformen:	Beamer und Tafel/Whiteboard, Simulationsprogramme, elektronische Skripten und Arbeitsunterlagen, praktische Übungen.
Literatur:	<p>Günther Wellenreuther / Dieter Zastrow: Automatisieren mit SPS, Vieweg Verlag Wiesbaden 4. Auflage 2008, EAN 978-3-8348-0231-6</p> <p>Karl-Heinz John, Michael Tiegelkamp, SPS-Programmierung mit IEC 61131-3, Konzepte und Programmiersprachen, Anforderungen an Programmiersysteme, Entscheidungshilfen. VDI-Buch,</p>

	<p>Springer-Verlag 4. Auflage 2009, EAN 978-3-6420-0268-7</p> <p>Eberhardt Grötsch, SPS - Speicherprogrammierbare Steuerungen, Oldenbourg Verlag München 5. Auflage 2004, EAN 978-3-8356-7043-3</p> <p>Raimond Pigan, Mark Metter (Absolvent unserer Fakultät), Automatisieren mit PROFINET: Industrielle Kommunikation auf Basis von Industrial Ethernet, Publicis Corporate Publishing Erlangen, 2. Auflage 2008</p> <p>Michael Braun, Objektorientiertes Programmieren, Grundlagen, Programmierbeispiele und Softwarekonzept nach IEC61131-3, Publicis Pxelpark Erlangen, 2016, ISBN 978-3-89578-455-2.</p> <p>PLCopen: www.plcopen.org</p> <p>sowie weitere Bücher und URL Links</p>
--	---

Modulbezeichnung	Verteilte Systeme
Kürzel	VS
Lehrform / SWS	Seminaristischer Unterricht / 2 SWS; Praktikum / 2 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60 Präsenzstunden; 90 Stunden Eigenarbeit
Fachsemester	6 oder 7
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Dieter Wißmann
Dozent(in)	Prof. Dr. Dieter Wißmann
Sprache	Deutsch
Zuordnung zum Curriculum	Fachwissenschaftliches Wahlpflichtfachmodul
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5
Inhaltliche Voraussetzungen	Kenntnisse in den Programmiersprachen C und Java; Grundkenntnisse in Betriebssystemen und Datennetzen.
Qualifikationsziele, Kompetenzen	<p>Fachlich-methodische Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden sollen die Unterschiede zwischen verteilten Systemen und zentralistischen Systemen kennen lernen und verstehen. Insbesondere soll verinnerlicht werden, welche zusätzlichen Probleme bei verteilten Systemen auftreten. • Sie sollen Kenntnisse erwerben, wie die zusätzlichen Probleme durch prinzipiell neue Konzepte und Algorithmen gelöst werden können. • Sie sollen die Fähigkeit erlangen, die Architektur von verteilten Systemen einzuordnen, verteilte Systeme zu entwerfen und mit Hilfe von etablierten Mechanismen/Ansätzen zu implementieren. • Es soll ein Verständnis erworben werden, welche Basisdienste in verteilten Systemen notwendig sind.

Lehrinhalt	<ul style="list-style-type: none"> • Klassifikation und Architektur von verteilten Systemen • Parallelität und Konkurrenz <ul style="list-style-type: none"> ◦ Threads und Threadsynchroisation • Client-Server-Kommunikation <ul style="list-style-type: none"> ◦ Nachrichtenbasierte Koordination ◦ Sockets • Diverse Middleware-Mechanismen/Ansätze <ul style="list-style-type: none"> ◦ RPC, RMI, CORBA ◦ Webservices, REST • Basisdienste in verteilten Systemen
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min) und praktische Leistungsnachweise im Rahmen einer praktische Studienarbeit im Verhältnis 1:1
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Overhead; Lernmanagementsysteme (Moodle), Elektronisches Skript und Arbeitsunterlagen; PC-Systeme;
Literatur	Abts D.: Masterkurs Client/Server-Programmierung mit Java, 4. Auflage; Vieweg, 2015. Bengel G.: Verteilte Systeme, 4. Auflage; Springer Vieweg, Wiesbaden, 2014. Coulouris G., Dollimore J., Kindberg T., Blair G.: Distributed Systems, Concepts and Design; 5. Auflage; Pearson, 2012. Tanenbaum A., van Steen M.: Distributed Systems, Principles and Paradigms; 2. Auflage, Prentice Hall Pearson, 2014.

3.5 Abschlussarbeit

Modulbezeichnung	Bachelorarbeit
Kürzel	-
Lehrform / SWS	0 SWS
Leistungspunkte	12 ECTS
Arbeitsaufwand	360 h Eigenarbeit
Fachsemester	7
Angebotsturnus	halbjährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Alle Professoren der Informatik
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §8 Abs. 2 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Ziele: Studierende sollen</p> <ul style="list-style-type: none"> das im Studium erworbene Wissen auf die Fragestellung einer wissenschaftlichen Bachelorarbeits-Fragestellung anwenden und analysieren können, eine wissenschaftliche Ausarbeitung auf der Grundlage wissenschaftliches Arbeiten selbstständig verfassen können. <p>Persönliche-Soziale Kompetenzen: Studierende sind in der Lage:</p> <ul style="list-style-type: none"> sich an der Diskussion fachlich-methodischer Probleme mit dem Lehrpersonal, mit anderen Studierenden und Personen aus der beruflichen Praxis zu beteiligen,

	<ul style="list-style-type: none"> • anwendungsorientierte Bachelorarbeits-Fragestellungen selbstständig zu erschließen, • ihren wissenschaftlichen Standpunkt zu verschriftlichen, andere publizierte Meinungen zu bewerten, zu respektieren und die daraus gewonnenen Erkenntnisse in die Bachelor Arbeit schriftlich einzuarbeiten.
Lehrinhalte	Abhängig vom Thema der Bachelorarbeit
Endnotenbildende Studien- / Prüfungsleistungen	Bachelorarbeit
Sonstige Leistungsnachweise	-
Medienformen	-
Literatur	H. Balzert, M. Schröder, C. Schäfer: Wissenschaftliches Arbeiten. W3L-Verlag, Dortmund, in der neuesten Auflage und andere Spezialliteratur

Modulbezeichnung	Bachelorseminar
Kürzel	Bcsem
Lehrform / SWS	1 SWS
Leistungspunkte	3 ECTS
Arbeitsaufwand	15 h Präsenz (Seminarpräsentationen) 75 h Eigenarbeit (Seminarvorbereitung)
Fachsemester	7
Angebotsturnus	halbjährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Alle Professoren der Informatik
Sprache	Deutsch
Nutzung in anderen Studiengängen	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §8 Abs. 2 SPO
Inhaltliche Voraussetzungen	-
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Ziele: Die Studierenden können</p> <ul style="list-style-type: none"> • Zielsetzungen und Fortschritte ihrer Bachelorarbeit herausarbeiten und präsentieren, • die präsentierten Inhalte aus anderen Bachelorinhalten kritisch hinterfragen und würdigen, • die Inhalte Ihrer Bachelorarbeit verteidigen. <p>Persönliche-Soziale Kompetenzen: Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • ihr Bachelor-Thema wissenschaftlich zu präsentieren, • sich an der Diskussion fachlich-methodischer Fragestellungen im Seminar mit dem Lehrpersonal, anderer Studierender und Personen aus der beruflichen Praxis zu beteiligen,

	<ul style="list-style-type: none"> • ihren Standpunkt mündlich zu vertreten, andere Meinungen in der Diskussion zu respektieren und die daraus gewonnenen Erkenntnisse zu reflektieren, • Zeitressourcen einzuteilen.
Lehrinhalte	Abhängig von den Themen der Bachelorarbeiten
Endnotenbildende Studien- / Prüfungsleistungen	Präsentation
Sonstige Leistungsnachweise	-
Medienformen	Beamer und Tafel/Whiteboard
Literatur	H. Balzert, M. Schröder, C. Schäfer: Wissenschaftliches Arbeiten. W3L-Verlag, Dortmund, in der neusten Auflage und andere Spezialliteratur