



HOCHSCHULE | FAKULTÄT
COBURG | Elektrotechnik und Informatik

Modulhandbuch

BACHELORSTUDIENGANG VISUAL COMPUTING (VC)

Studienverlauf Visual Computing (B.Sc.)

1	2	3	4	5	6	7
Analysis (7)	Diskrete Mathematik (5)	Algorithmen und Datenstrukturen (5)	WPF Informatik (5)	Praxissemester (30)	Interdisziplinäre Projektarbeit (10)	Bachelorarbeit mit Seminar (15)
	Grundlagen der Informatik (7)	Webtechnologien (7)	Fortgeschrittene Programmierung (5)			
Mathematik für Visual Computing (5)			Bildverarbeitung 2 (5)			
Programmieren 1 (5)	Programmieren 2 (7)	Computergrafik 1 (5)	Computergrafik 2 (5)		WPF VC (5)	WPF VC (5)
Grundlagen des Visual Computing (7)		Mensch-Maschine-Interaktion 1 (5)	Bildverarbeitung 1 (5)		VC Seminar (3)	WPF VC (5)
	Künstliche Intelligenz 1 (6)	WIA (3)			Mensch-Maschine-Interaktion 2 (5)	WPF Informatik (5)
WPF Schlüsselqualifikation (4)			Englisch (4)			

WIA = Wissenschaftliches und interdisziplinäres Arbeiten
 WPF = Wahlpflichtfach
 VC = Visual Computing
 Zahlen in Klammern = ECTS Punkte

■ Informatik ■ Interdisziplinär
■ Visual Computing ■ Praxis

 **COBURG UNIVERSITY**
of applied sciences and arts
Stand: 06.02.2023

ECTS = European Credit Transfer System

Zahlen in Klammern = ECTS Punkte (in Summe ergeben sich 210 ECTS-Punkte)

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, tiefgehende 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			
1.1 Allgemeine Informatik Pflichtmodule			
Analysis	x		
Diskrete Mathematik	x		
Grundlagen der Informatik	x		
Programmieren 1	x		
Programmieren 2	x		
Webtechnologien	x		
1.2 Visual Computing Pflichtmodule			
Grundlagen des Visual Computing	x		
Mensch-Maschine-Interaktion 1	x		
Künstliche Intelligenz 1	x		
1.3 Schlüsselqualifikationen			
Wahlpflichtmodul Schlüsselqualifikation: Siehe Fächer des Studium Generale			
2. Erster Studienabschnitt – theoretische Studiensemester 3 und 4			
2.1 Allgemeine Informatik Pflichtmodule			
Algorithmen und Datenstruktur	x		
Fortgeschrittene Programmierung	x		
2.2 Visual Computing Pflichtmodule			
Computergrafik 1	x		
Computergrafik 2	x		
Bildverarbeitung 1	x		
Bildverarbeitung 2	x		
Künstliche Intelligenz 2	x		
Mathematik für Visual Computing	x		
Mensch-Maschine-Interaktion 2	x		
Visual-Computing-Seminar	x		
2.3 Allgemeine Informatik Wahlpflichtmodule			
Betriebssysteme	x		
Computernetze	x		
IT-Sicherheit	x		
Shell und Prozesse	x		
Software-Anforderungen und -Modellierung	x		
Software-Architekturen und -Testen	x		
Stochastik	x		
2.4 Schlüsselqualifikationen			
Englisch (GER B2) 2	x		
Englisch (GER B2) 1	x		
Wissenschaftliches und interdisziplinäres Arbeiten	x		

	Gefährdung Mutterschutz		
	Grün	Gelb	Rot
3. Dritter Studienabschnitt – theoretische Studiensemester 5 und 7			
3.1 Praktisches Studiensemester			
Industriepraktikum	x		
Praxisbegleitende Lehrveranstaltungen	x		
Praxisseminar	x		
3.2 Fachwissenschaftliches Pflichtmodul			
Interdisziplinäre Projektarbeit	x		
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		
3.4 Fachwissenschaftliche Wahlpflichtmodule Informatik			
Betriebssysteme	x		
Computernetze	x		
Digitale Systemintegration	x		
Eingebettete Betriebssysteme	x		
E-Entrepreneurship	x		
Grundlagen der Wirtschaftsinformatik	x		
IT-Sicherheit	x		
Mikrocomputertechnik	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	8
1.1.1 Fachwissenschaftliche Pflichtmodule Informatik.....	8
Analysis	8
Diskrete Mathematik	10
Grundlagen der Informatik	12
Programmieren 1	14
Programmieren 2	16
Webtechnologien.....	18
1.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing	21
Grundlagen des Visual Computing.....	21
Künstliche Intelligenz 1	23
Mensch-Maschine-Interaktion 1	25
1.2 Schlüsselqualifikationen (SQF).....	27
Wahlpflichtmodule Schlüsselqualifikationen.....	27
2. Erster Studienabschnitt – theoretische Studiensemester 3 und 4	28
2.1.1 Fachwissenschaftliche Pflichtmodule Informatik.....	28
Algorithmen und Datenstrukturen	28
Fortgeschrittene Programmierung.....	31
2.1.2 Fachwissenschaftliche Pflichtmodule Visual Computing	33
Bildverarbeitung 1.....	33
Bildverarbeitung 2.....	35
Computergrafik 1	37
Computergrafik 2	39
Künstliche Intelligenz 2	41
Mathematik für Visual Computing	43
Mensch-Maschine-Interaktion 2.....	45
Visual-Computing-Seminar	47
2.2 Allgemeine Informatik Wahlpflichtmodule	49
Betriebssysteme	49
Computernetze.....	51
IT-Sicherheit.....	53
Shell und Prozesse	55
Software-Anforderungen und -Modellierung.....	58
Software-Architekturen und -Testen.....	61
Stochastik	64

2.4 Schlüsselqualifikationen	66
Englisch (GER B2) 1	66
Englisch (GER B2) 2	68
Wissenschaftliches und interdisziplinäres Arbeiten.....	70
3. Dritter Studienabschnitt – Studiensemester 5 bis 7	72
3.1 Praktisches Studiensemester	72
Industriepraktikum	72
Praxisbegleitende Lehrveranstaltungen	74
Praxisseminar	75
3.2 Fachwissenschaftliches Pflichtmodul	77
Interdisziplinäre Projektarbeit.....	77
3.3 Fachwissenschaftliche Wahlpflichtmodule Visual Computing	79
Bildanalyse und Bildsynthese	79
3D Visualisierung	81
Evaluierung von Benutzerschnittstellen	83
GPU Image Synthesis	85
Geometrieverarbeitung	87
Grundlagen Game Design.....	89
Grundlagen des Maschinellen Lernen	91
Grundlagen der Datenkompression	93
3.4 Fachwissenschaftliche Wahlpflichtmodule Informatik	95
Betriebssysteme	95
Computernetze.....	97
Digitale Systemintegration.....	99
Eingebettete Betriebssysteme	101
E-Entrepreneurship	104
Grundlagen der Wirtschaftsinformatik.....	104
IT-Sicherheit.....	108
Mikrocomputertechnik.....	110
Rechnerarchitekturen	114
Robotik	117
SAP-Systeme – Schnittstellen und ABAP-Programmierung.....	119
Secure Software Engineering.....	121
Shell und Prozesse	123
Software-Anforderungen und -Modellierung.....	126
Software-Architekturen und -Testen.....	129
Software Engineering.....	132

Softwareentwurf in der Automatisierungstechnik.....	134
Stochastik.....	137
Verteilte Systeme.....	139
3.5 Abschlussarbeit.....	141
Bachelorarbeit.....	141
Bachelorseminar.....	143

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 Differentialrechnung Extremwerte, Taylorentwicklung
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (120 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Tafel, Skript
Literatur	T. Arens et al., „Mathematik“, Spektrum, Heidelberg, 2008

	<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 grundlegenden Algorithmen kennen und anwenden können.
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. • Fischer & Springborn: Lineare Algebra: Eine Einführung für Studienanfänger. 19. Auflage, Springer, 2020.

	<ul style="list-style-type: none">• 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	<p>Fachlich-methodische Kompetenzen:</p> <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. <p>Persönliche-Soziale Kompetenzen:</p> <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

	<ul style="list-style-type: none"> • Daten und Information • Zahlendarstellung • Boolesche Algebra <p>Aufbau und Funktionsweise von Rechnern Von-Neumann-Rechner Hardware (Hauptplatine, Peripherie, CPU, Busse) Leistungsmerkmale, Einsatzbereiche Registermaschine</p> <p>Konzepte von Programmiersprachen Maschinencode, Assembler-, Hochsprache Datentypen Prozedurale Abstraktion Input/Output, Kontrollstrukturen Pseudocode, Strukto-, Flussdiagramm</p> <p>Komplexität von Algorithmen Datenstrukturen + Suchen Felder, Warteschlange, Stack, Liste, Graph, Baum</p> <p>Algorithmenentwurf Entwurfstechniken, Induktion, Rekursion Sortieralgorithmen</p> <p>Softwareentwicklung Vorgehensmodelle, Werkzeuge Compiler, Linker, Interpreter, Debugger</p> <p>Automatentheorie DEA, Mealy- und Moore-Automaten Akzeptierte Sprache Minimale und Kellerautomaten Turing-Maschine</p> <p>Informations- und Codierungstheorie Informationsquellen, Informationsgehalt, Entropie Shannon-Fano- und Huffman-Codierung Fehlererkennung, Paritätsbit, Hamming-Distanz, Hamming-Code</p>
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</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: 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: 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

	<p>bearbeiten und Lösungsvarianten zu reflektieren,</p> <ul style="list-style-type: none"> • 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	Informatik Bc
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	Webtechnologien
Kürzel	Wt
Lehrform / SWS	6 SWS
Leistungspunkte	7 ECTS
Arbeitsaufwand	210 h, davon <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 (2021): JavaScript. Das umfassende Handbuch, 3. Aufl., Rheinwerk Computing, Bonn. • Müller, P. (2020): Einstieg in HTML und CSS – Webseiten programmieren und gestalten mit HTML und CSS – ganz ohne Vorwissen, Rheinwerk Computing, Bonn.

	<ul style="list-style-type: none">• Müller, P. (2015): Flexible Boxes – Eine Einführung in moderne Websites; [alle wichtigen HTML5-Elemente und CSS3-Eigenschaften ; Grundlagen und Konzepte für Responsive Webdesign ; responsive Grafiken, mobile Navigation, Gridlayouts und Flexbox], 2. Aufl., Rheinwerk Computing, Bonn.• Wolf, J. (2019): HTML5 und CSS3: das umfassende Handbuch, 3. 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	1
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

	<ul style="list-style-type: none"> • 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>

Modulbezeichnung	Künstliche Intelligenz 1
Kürzel	KI1
Lehrform / SWS	4 SWS
Leistungspunkte	6 ECTS
Arbeitsaufwand	180h
Fachsemester	2
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	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"> ○ Tiefen- und Breitensuche ○ A*, RBFS (kürzeste Route) ○ Suche im kontinuierlichen Suchraum ○ Suche mit Nicht-Determinismus ○ Suche mit versteckten Zuständen • Suche in Spielen (mit Gegnern) <ul style="list-style-type: none"> ○ Alpha-Beta-Pruning ○ Heuristische Echtzeit-Entscheidungen ○ 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	Mensch-Maschine-Interaktion 1
Kürzel	MMI1
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	2
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>

1.2 Schlüsselqualifikationen (SQF)

Wahlpflichtmodule Schlüsselqualifikationen

Studium Generale Kurse, die im ersten Semester als Schlüsselqualifikation von Informatik und Visual Computing Studierenden belegt werden können:

Zeit- und Lernmanagement
Schutzfaktor Resilienz - Kompetenz zur Stressbewältigung entwickeln
Visionen entwickeln und Ziele erreichen
Stressmanagement nach dem BERN Prinzip
Starke Mitte - Mentaltraining für Studium und Leben
Gesundheitskompetenz im Studium und Alltag
"Resilient und selbstbewusst" studieren
Gelassenheit im (Studien-)Alltag

Es sollte ein Modul aus dieser Liste gewählt werden. Das zweite SQF-Modul kann aus dem weiteren Angebot des Studium Generale inkl. Sprachen gewählt werden.

Weiter Informationen zu den einzelnen Modulen sowie den Anmeldemodalitäten finden Sie unter folgendem Link des Studium Generale:

<https://mycampus.hs-coburg.de/de/service-offer/studium-generale-wintersemester-202324-uebersicht-und-info>

Bei Fragen zu den Kursen, wenden Sie sich bitte an wiku@hs-coburg.de.

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	<p>Lang, H.W.: „Algorithmen in Java“, Oldenbourg Verlag jeweils in der neusten Auflage</p> <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. Thomas Wieland
Dozent(in)	Prof. Dr. Thomas Wieland
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, Initialisierung mit Listen, Kopierkonstruktor,

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

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. oder 7. Semester
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Höhere Mathematik, Kenntnisse 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</p>

	<p>technischen 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 • Segmentieren
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	<p>Studierende lernen Werkzeuge und Vorgehensweisen kennen um fortgeschrittene Systeme der Bildverarbeitung und Computer Vision entwerfen und entwickeln zu können. Lehrinhalte umfassen:</p> <ul style="list-style-type: none"> • Projektive Geometrie • Robuste Parameterschätzung • Maschinelles Lernen für Computer Vision

	<ul style="list-style-type: none"> • 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 30h) 90h Eigenarbeit (Seminaristischer Unterricht 45h, Übung 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"> • Geometrierepräsentationen • Rastergraphik und Scankonvertierung

	<ul style="list-style-type: none"> • Graphik-Pipeline • Graphik-Hardware • Clipping • Transformationen (Perspektive und Projektionen) • Sichtbarkeitsbestimmung • Schattierung • Lokale Beleuchtungsmodelle • Texturen • WebGL-Graphik API
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> <p>Kessenich, Sellers, Shreiner: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V, Addison Wesley, 2016</p>

Modulbezeichnung	Computergrafik 2
Kürzel	CG2
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	60h Präsenz (Seminaristischer Unterricht 30h, Praktikum 30h) 90h Eigenarbeit (Praktikum, Erstellung der praktischen Studienarbeit)
Fachsemester	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Quirin Meyer
Dozent(in)	Prof. Dr. Quirin Meyer / Prof. Dr. habil. Jens Grubert
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 • Lösungsvorschläge von Programmier- und Theorieaufgaben vor einer Gruppe von Studierenden präsentieren und diskutieren

Lehrinhalte	<ul style="list-style-type: none"> • Texturdarstellung • MIP-Mapping • Partielle Ableitung von multivariaten Funktionen und deren Approximation • Height Maps, Normal Maps, Shadow Maps, Light Maps • Alpha-Blending • Anti-Aliasing Verfahren • Ray-Tracing • Beschleunigungsstrukturen • Datenstrukturen zur Geometrie- und Szenerepräsentation
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung 90 Minuten
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>

Modulbezeichnung	Künstliche Intelligenz 2
Kürzel	KI 2
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 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. • einfache Probleme durch Aussagenlogik und Prädikatenlogik darstellen und algorithmisch lösen können. • Grundkenntnisse Bayesscher Wahrscheinlichkeitslehre verstehen und rationale Agenten zur Lösung von Problemen mit Wahrscheinlichkeiten programmieren können. • Algorithmen zum Lernen durch Beispiele kennen und verstehen und zur Problemlösung 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"> • Logische Agenten <ul style="list-style-type: none"> ○ Aussagenlogik und Prädikatenlogik 1. Stufe ○ Wissensrepräsentation durch Logik ○ Forward / Backward Chaining und Resolution • Probabilistisches Schließen <ul style="list-style-type: none"> ○ Bayessche Wahrscheinlichkeit ○ Hidden Markov Models • Lernen durch Beispiele / Maschinelles Lernen
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	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 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"> ○ Methode der kleinsten Quadrate ○ Lineare Optimierung, lineare Ausgleichsrechnung ○ Nicht-lineare Optimierung • Multivariate Analysis (Kurven, Flächen)

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

Modulbezeichnung	Mensch-Maschine-Interaktion 2
Kürzel	MMI2
Lehrform / SWS	4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150h
Fachsemester	4
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) • 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., & Dachsel, R. (2010). Interaktive systeme: band 1: grundlagen, graphical user interfaces, informationsvisualisierung. Springer-Verlag.</p> <p>Preim, B., & Dachsel, 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: Springer-Verlag.</p>

Modulbezeichnung	Visual-Computing-Seminar
Kürzel	VCSem
Lehrform / SWS	2 SWS
Leistungspunkte	3 ECTS
Arbeitsaufwand	30 h Präsenz (Seminaristischer Unterricht, Seminarvorträge) 60 h Eigenarbeit (Vorbereitung Präsentation / Hausarbeit)
Fachsemester	4
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 §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Kenntnisse von Grundkonzepten der Informatik
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Ziele: Die Studierenden können</p> <ul style="list-style-type: none"> • sich in ein vorgegebenes Informatik-Fachthema einarbeiten, • selbständig unter Anleitung nach geeigneten Literaturquellen recherchieren, • den Inhalt eines Themas in Form einer an ein wissenschaftliches Paper angelehnten schriftlichen Ausarbeitung verfassen <p>Persönliche-Soziale Kompetenzen: Studierende sind in der Lage:</p> <ul style="list-style-type: none"> • ihr Fachthema mündlich im Plenum zu präsentieren und zu diskutieren • sich verbal und schriftlich wissenschaftlich auszudrücken • sich kritisch mit dem eigenen Thema und mit Fachthemen anderer Studierender wissenschaftlich

	und gesellschaftlich auseinanderzusetzen und diese zu reflektieren
Lehrinhalte	<ul style="list-style-type: none"> • Hinweise zum wissenschaftlichen Arbeiten • Individuelle Einarbeitung • Präsentationen der Seminarthemen inklusive Diskussion
Endnotenbildende Studien- / Prüfungsleistungen	Hausarbeit und Präsentation
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel
Literatur	Abhängig vom Projektthema sowie H. Balzert, M. Schröder, C. Schäfer: Wissenschaftliches Arbeiten. W3L-Verlag, Dortmund, 2011

2.2 Allgemeine Informatik Wahlpflichtmodule

	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	4
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"> ○ Schutzmechanismen ○ Deadlockerkennung • Arbeitsspeicherverwaltung <ul style="list-style-type: none"> ○ Adressierungsmodelle, Adressräume ○ 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	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	4 und 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	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
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	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, Informatik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 SPO
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 die Grundlagen der IT-Sicherheit verstehen, die wichtigsten Risiken für diese Grundwerte verstehen und in vorbereiteten Szenarien anwenden können sowie ausgewählte Maßnahmen und Techniken zur Vermeidung dieser Risiken verstehen und in praktischen Übungen anwenden lernen.
Lehrinhalte	Klassische Verschlüsselungsverfahren Grundbegriffe der Kryptographie

	<p>Endliche Zahlenmengen und Restklassen</p> <p>Blockchiffren</p> <p>Rechnung mit Potenzen modulo n</p> <p>Grundlagen Kryptoanalyse</p> <p>Public-Key-Kryptographie</p> <p>Elliptische Kurven Kryptographie</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>PC-Übung mit virtualisierter Übungsumgebung</p>
Literatur	<ul style="list-style-type: none"> ● J. Swoboda, S. Spitz, M. Pramateftakis: Kryptographie und IT-Sicherheit. Vieweg Studium, 2008, 39,95 € ● J. Buchmann: Einführung in die Kryptographie. Springer Verlag, 5. Auflage, 2010 ● C. Eckert: IT-Sicherheit. Oldenbourg-Verlag, 2009 ● B. Schneier: Angewandte Kryptographie. Pearson Studium, 2008 ● W. Ertel: Angewandte Kryptographie. Hanser Verlag, 3. Aufl., 2007. ● A. Menezes, P. Oorschot und S. Vanstone. CRC Press, Handbook of Applied Cryptography, 5. Aufl. 2001.

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	4 oder 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"> ○ Automatisierung von Kommandofolgen ○ Variablen ○ Fallunterscheidungen, Schleifen ○ Verarbeitung von Zeichenketten • Prozesse in Linux <ul style="list-style-type: none"> ○ Erzeugung und Beendigung ○ Wertübergabe und Wertrückgabe ○ Signale zum Steuern von Prozessen ○ Vordergrund- und Hintergrundprozesse • Interprozesskommunikation in Linux <ul style="list-style-type: none"> ○ Umlenkung von Kanälen/Pipes (in der Shell) ○ Semaphoren (im C-Programm, System-API) ○ Shared Memory (im C-Programm, System-API) • Entwicklungswerkzeuge <ul style="list-style-type: none"> ○ Generierung ausführbarer Code (C-Compiler) ○ Fehlersuche
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min) und praktische Leistungsnachweise im Rahmen einer praktische Studienarbeit im Verhältnis 1:1
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel; Lernmanagementsysteme (Moodle), Elektronisches Skript und elektronische Arbeitsunterlagen; PC-Systeme;
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, 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	4 oder 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 Industrewirtschaft
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
Inhaltliche Voraussetzungen	Grundlagen Software Engineering
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: Studierende sollen <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

	<ul style="list-style-type: none"> • die Rolle von Geschäftsprozessen als Anforderungsquelle verstehen, • 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>

	Weiterführende Spezialliteratur für die verschiedenen Kapitel
--	---------------------------------------------------------------

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	4 oder 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	Fachlich-methodische Kompetenzen: Studierende sollen <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

	<p>Artefakte (z.B. Anforderungen, Code, Modelle) kennen, verstehen und anwenden können.</p> <p>Persönliche-Soziale Kompetenzen: 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	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	4 + 6
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.

2.4 Schlüsselqualifikationen

Modulbezeichnung	Englisch (GER B2) 1
Kürzel	Eng1
Lehrform / SWS	2 SWS
Leistungspunkte	2 ECTS
Arbeitsaufwand	30 h Präsenz, 30 h Eigenarbeit
Fachsemester	3
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	B. Craven, M.A.
Dozent(in)	B. Craven, M.A. / 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 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) oder Präsentation (20 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	4
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	B. Craven, M.A.
Dozent(in)	B. Craven, M.A. / 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) oder Präsentation (20 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Visualizer
Literatur	Skript

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	3
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	Interdisziplinäre Projektarbeit
Kürzel	IntProa
Lehrform / SWS	Projekt / 30h
Leistungspunkte	10 ECTS
Arbeitsaufwand	30h Präsenzstudium, 270h Eigenarbeit
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Meyer
Dozent(in)	Dozenten des Visual Computing
Sprache	Deutsch
Nutzung in anderen Studiengängen	
Zulassungsvoraussetzungen	Vorrückensberechtigung in den dritten Studienabschnitt gemäß SPO B VC vom 29.07.2020, §5 Abs. 2.
Inhaltliche Voraussetzungen	
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • ausgewählte Techniken aus den Bereichen Anforderungsanalyse, Software-Architektur, Software-Entwicklung, Software-Testen, Projektmanagement auf einer komplexen, interdisziplinären Problemstellung in einem Anwendungsfeld des Visual Computing verwenden, • notwendige Werkzeuge erlernen und anwenden können <p>Persönliche-Soziale Kompetenzen:</p> <ul style="list-style-type: none"> • sich als Team selbst organisieren und erfolgreich arbeiten können • ihre Kommunikations- und Teamfähigkeit stärken,

Lehrinhalte	<ul style="list-style-type: none"> • Werkzeuge, Methoden und Techniken der Software-Entwicklung und Software-Architektur-Entwurfs • Realisierung einer komplexen, interdisziplinären Software-Aufgabenstellung im Team von bis zu 5 Personen
Endnotenbildende Studien- / Prüfungsleistungen	Praktische Studienarbeit
Sonstige Leistungsnachweise	
Medienformen	
Literatur	<p>Pichler, R.: SCRUM – Agiles Projektmanagement erfolgreich einsetzen. dpunkt-Verlag, jeweils in der aktuellen Auflage</p> <p>Arlow, J.; Neustadt, I.: UML2 and the Unified Process. Addison-Wesley, jeweils in der neusten Auflage</p> <p>Bass, L., Kazman, R., Clements, P.: Software Architecture in Practices, Addison Weseley, jeweils in der neusten Auflage</p>

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: 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: 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"> ○ Grunddimensionen Farbe und Tiefe ○ Strukturen, Formen, Objekte ○ Objektdetektion ○ Gesichtsdetektion ○ Dimension Zeit (Bewegung)

	<ul style="list-style-type: none"> • Grundlagen Bildsynthese <ul style="list-style-type: none"> ○ Modifikation ○ Kreation ○ Interaktion ○ 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 • <u>Birchfield</u>: 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).
Lehrinhalte	<ul style="list-style-type: none"> • Begriffe, Abgrenzungen zu anderen Konzepten • Visualisierungsprozess • Datencharakterisierung • Einflussfaktoren

	<ul style="list-style-type: none"> • 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>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: Studierende sollen:</p> <ul style="list-style-type: none"> • verschiedene Evaluierungsmethoden kennen und praktisch anwenden können • Usability-Evaluation durchführen und planen können • Qualitative und quantitative Daten erheben und auswerten können <p>Persönliche-Soziale Kompetenzen: 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 • Entwicklungsschritte zu dokumentieren
Lehrinhalte	Arten der Evaluation

	<ul style="list-style-type: none"> - Feldstudien und Laborexperimente, Onlinestudien, empirische und analytische Evaluation, qualitative und quantitative Evaluation <p>Psychologische Grundlagen für die Systementwicklung</p> <ul style="list-style-type: none"> - Kognitive Aspekte (Wahrnehmung, Aufmerksamkeit, Wissensrepräsentationen, Gedächtnis) - Arbeitspsychologische Aspekte <p>Qualitative Methoden:</p> <ul style="list-style-type: none"> - Fragebögen - Cognitive Walkthrough - Heuristische Evaluation - Usability Tests <p>Quantitative Methoden:</p> <ul style="list-style-type: none"> - Beobachtungsstudien - Experimente - Modellbasierte Evaluation <p>Evaluation von erweiterten Benutzerschnittstellen (3D Interfaces, VR / AR, multimodale Interaktionen)</p>
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	Glms
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: 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: 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 • Entwicklungsschritte zu dokumentieren

	<ul style="list-style-type: none"> • 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 Lernen
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. Florian Mittag
Dozent(in)	Prof. Dr. Florian Mittag
Sprache	Deutsch
Nutzung in anderen Studiengängen	Informatik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	lineare Algebra
Qualifikationsziele / Kompetenzen	<p>Fachlich-methodische Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • die grundlegenden Prinzipien und Methoden des Maschinellen Lernens kennen, verstehen und anwenden können. • die Funktionsweise gängiger Algorithmen des Maschinellen Lernen kennen, verstehen und einsetzen können. • unbekannte reale Datensätze mittels Maschinellem Lernverfahren analysieren und wissenschaftliche fundierte Auswertungen erstellen können. <p>Persönliche-Soziale Kompetenzen: Studierende sollen</p> <ul style="list-style-type: none"> • gesellschaftliche Auswirkung des verstärkten Einsatzes des Maschinellen Lernens analysieren und bewerten können. • die moralischen und ethischen Aspekte des Maschinellen Lernens analysieren und bewerten können.

Lehrinhalte	<ul style="list-style-type: none"> • Was ist Maschinelles Lernen? • Supervised Learning: Klassifikation und Regression <ul style="list-style-type: none"> ○ Güte von Klassifizierern ○ Feature Selection/Engineering ○ Hyperparameter Selection ○ Support Vector Machines ○ Künstliche Neuronale Netze • Evolutionäre Algorithmen • Reinforcement Learning • Gängige Herausforderungen bei der Anwendung
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten)
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel, Lernmanagementsysteme (Moodle), praktische Übungen
Literatur	Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag, 2006

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	6-7
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 (Laufängen-, 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	• Strutz: Bilddatenkompression, 5.Auflage oder höher
-----------	------------------------------------------------------

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	Fachlich-methodische Kompetenzen: <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

	<ul style="list-style-type: none"> • Synchronisation von Prozessen und Threads <ul style="list-style-type: none"> ○ Schutzmechanismen ○ Deadlockerkennung • Arbeitsspeicherverwaltung <ul style="list-style-type: none"> ○ Adressierungsmodelle, Adressräume ○ 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	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	4 und 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	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 1 SPO
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	Digitale Systemintegration
Kürzel	DSi
Lehrform / SWS	Seminaristischer Unterricht (2 SWS), Praktikum (2 SWS) / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	Präsenzzeit: 60h, Selbststudium: 90h
Fachsemester	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Oliver Engel
Dozent(in)	Prof. Oliver Engel
Sprache	deutsch
Nutzung in anderen Studiengängen	Automatisierungstechnik und Robotik
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §6 Abs. 2 SPO
Inhaltliche Voraussetzungen	Grundlagen der Technischen Informatik, Grundlagen der Mikrocomputertechnik
Qualifikationsziele / Kompetenzen	<ol style="list-style-type: none"> 1. Studierende erlangen die Fähigkeit, komplexe Systemanforderungen in ein integriertes System aufzuteilen und umzusetzen. Dabei können sie treffsicher die Zieltechnologien auswählen. 2. Sie beherrschen den Umgang mit Entwicklungsumgebungen für Hardware-/Software Codesign und können komplexe digitale Designs auf eine Zielhardware integrieren. 3. Studierende wissen, wie die Kommunikation unterschiedlicher Systemkomponenten sinnvoll aufgebaut werden kann.
Lehrinhalte	<p>CMOS-Technologie</p> <ul style="list-style-type: none"> • Grundlegende Eigenschaften: Leistungsverhalten, Laufzeit, Flächenverbrauch • Untersuchung von Fehlerursachen in komplexen Designs • Laufzeitoptimierung <p>Synchrones Design</p> <ul style="list-style-type: none"> • Design Rules zur Qualitätssicherung komplexer digitaler Schaltungen

	<p>Architekturen kundenspezifischer Digitalsysteme</p> <ul style="list-style-type: none"> • Programmierbare Logikbausteine: CPLD, FPGA • Kundenspezifische Hardware • On-Chip Bussysteme • Systemkomponenten: SRAM, DRAM <p>Test</p> <ul style="list-style-type: none"> • Fehlerarten • Testverfahren • Design for Testability
Endnotenbildende Studien- / Prüfungsleistungen	<p>Studienbegleitende schriftliche Prüfung: 60 Minuten</p> <p>Praktische Studienarbeit</p>
Sonstige Leistungsnachweise	keine
Medienformen	Tafel, Projektor, Entwicklungsumgebung
Literatur	<p>Göran Herrmann, Dietmar Müller: ASIC – Entwurf und Test, Fachbuchverlag Leipzig</p> <p>Ralf Gessler, Thomas Mahr: Hardware- Software- Codesign, Vieweg Verlag</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. Peter Raab
Dozent(in)	Prof. Dr. Peter Raab
Sprache	Deutsch
Nutzung in anderen Studiengängen	Bachelor-Studiengänge EL
Zulassungsvoraussetzungen	keine
Inhaltliche Voraussetzungen	Kenntnisse aus den Themenbereichen Rechnerarchitektur, maschinennahe Programmierung (C und Assembler) und Betriebssysteme
Qualifikationsziele / Kompetenzen	<p>Fachkompetenzen: Die Studierenden erlangen fundiertes Fachwissen ...</p> <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: 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: 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	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 Jochen Floherschütz
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	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. • Sie können die Herausforderungen bei Planung, Entwicklung/Beschaffung, Implementierung und Betrieb von Informations- bzw. Anwendungssystemen erklären

	<p>und sind in der Lage, entsprechende Beispiele/Aufgaben aus der Praxis zu bearbeiten.</p> <ul style="list-style-type: none"> • 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 • 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): Wirtschafts-informatik – 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, Visual Computing VC
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 SPO
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 die Grundlagen der IT-Sicherheit verstehen, die wichtigsten Risiken für diese Grundwerte verstehen und in vorbereiteten Szenarien anwenden können sowie ausgewählte Maßnahmen und Techniken zur Vermeidung dieser Risiken verstehen und in praktischen Übungen anwenden lernen.
Lehrinhalte	Klassische Verschlüsselungsverfahren Grundbegriffe der Kryptographie

	<p>Endliche Zahlenmengen und Restklassen</p> <p>Blockchiffren</p> <p>Rechnung mit Potenzen modulo n</p> <p>Grundlagen Kryptoanalyse</p> <p>Public-Key-Kryptographie</p> <p>Elliptische Kurven Kryptographie</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>PC-Übung mit virtualisierter Übungsumgebung</p>
Literatur	<ul style="list-style-type: none"> ● J. Swoboda, S. Spitz, M. Pramateftakis: Kryptographie und IT-Sicherheit. Vieweg Studium, 2008, 39,95 € ● J. Buchmann: Einführung in die Kryptographie. Springer Verlag, 5. Auflage, 2010 ● C. Eckert: IT-Sicherheit. Oldenbourg-Verlag, 2009 ● B. Schneier: Angewandte Kryptographie. Pearson Studium, 2008 ● W. Ertel: Angewandte Kryptographie. Hanser Verlag, 3. Aufl., 2007. ● A. Menezes, P. Oorschot und S. Vanstone. CRC Press, Handbook of Applied Cryptography, 5. Aufl. 2001.

Modulbezeichnung	Mikrocomputertechnik
Kürzel	Mct
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. Peter Johann Raab
Dozent(in)	Prof. Dr. Peter Johann Raab
Sprache	Deutsch
Nutzung in anderen Studiengängen	Automatisierungstechnik und Robotik Elektro- und Informationstechnik Energietechnik und Erneuerbare Energien
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §6 Abs. 1 SPO
Inhaltliche Voraussetzungen	Programmieren 1 und 2, Technische Informatik, Digitaltechnik
Qualifikationsziele / Kompetenzen	<p>Fachkompetenzen:</p> <p>Die Studierenden erlangen fundiertes fachliches Grundlagenwissen ...</p> <ul style="list-style-type: none"> - in der Anwendung von Embedded Systemen im industriellen Bereich: Sie erkennen die Abgrenzung zu klassischen Rechnersystemen und die Notwendigkeit von Embedded Systemen. - im Aufbau und von Eingebetteten Systemen: Sie erkennen die Strukturen von Hard- und Softwarekomponenten moderner Mikrocomputersystemen und Mikrocontrollern 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 eine maschinennahe Programmierung (Assembler) umsetzen.

	<p>Sie können Peripherieeinheiten des Mikrocontrollers in Betrieb nehmen und konfigurieren.</p> <ul style="list-style-type: none"> - in der Analyse und Umsetzung von Realzeiteigenschaften: Sie können asynchrone Ereignisse (Interrupts) erkennen und programmieren; 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, Oszi, Logikanalysator) 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 typische Modelle zur Beschreibung von Software (Flussdiagramm, Sequenzdiagramm, Zustandsdiagramm). - 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	Einführung: Begriffserklärung und Definitionen, Abgrenzung und Anforderungen von Eingebetteten Systemen, Anwendung von Mikrocontrollern

	<p>Rechnerarchitektur: Aufbau und Komponenten eines Mikrocontrollers (ARM-basiert), Hardware-Abstraktion (Programmiermodell), Befehlsverarbeitung, Funktionsweise des Rechnerkerns und der Peripherie, Speicherorganisation, Stack, Registersatz, Interruptverarbeitung</p> <p>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 (Embedded-C, Compiler)</p> <p>Peripherie eines Mikrocontrollers: Digitale Ein-/Ausgabe, Interrupts, Timer und Zähler, serielle Schnittstellen (UART, Bussysteme, Zugriffsverfahren), analoge Signale und Wandlung, externe Speicherschnittstellen</p> <p>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)</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung 90 Min. und prStA (Projekte und Testate)
Sonstige Leistungsnachweise	keine
Medienformen	Beamer / Präsentationsfolien, Vorlesungsskript, Übungsblätter mit Lösungen, Praktische Übungen am Rechner / Labor, Moodle
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> <p>NXP, UserManual LPC178x/7x User manual, UM10470, Rev. 4.0 — 21 December 2016</p> <p>NXP, Product Data Sheet LPC178x/7x 32-bit ARM Cortex-M3 microcontroller, Rev. 5.5 — 26 April 2016</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Modulbezeichnung	Rechnerarchitekturen
Kürzel	Ra
Lehrform / SWS	Seminaristischer Unterricht, Übung / 4 SWS
Leistungspunkte	5 ECTS
Arbeitsaufwand	150 h, davon <ul style="list-style-type: none"> • 90 h Präsenz (60 h Seminaristischer Unterricht, 30 h Übungen) • 60 h Eigenarbeit (20 h Nachbereitung des Lehrstoffs, 20 h Bearbeitung von Übungsaufgaben, 20 h Prüfungsvorbereitung)
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	-
Zulassungsvoraussetzungen	Vorrückensberechtigung nach §5 Abs. 2 SPO
Inhaltliche Voraussetzungen	Keine
Qualifikationsziele / Kompetenzen	<ol style="list-style-type: none"> 1. Studierende sollen die Fähigkeit erlangen, Rechnerkonzepte in Hard- und Software als Gesamtsystem zu durchdringen. 2. Durch Verständnis von logischen Funktionen sowie von elektronischen Verknüpfungsgliedern können Studierende Funktionen in digitale Hardware umzusetzen. 3. Studierende verstehen Codierungen, Zahlendarstellungen und grundlegende Algorithmen der Datenverarbeitung und können angepasste Rechensysteme entwerfen.
Lehrinhalte	Axiome der Boolschen Algebra, Normalformen (DNF, KNF) Minimierung von kombinatorischen Schaltungen mit Karnaugh-Veitch, Quine McCluskey Digitale Hardware: MOS-Transistor, Grundgatter der Digitaltechnik, Komplexgatter Standardschaltnetze: Codierer, Decodierer, Multiplexer, Komparatoren, Addierer, Carry Look Ahead

	<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>
Medienformen	Beamer / Präsentationsfolien, Vorlesungsskript, Übungsblätter mit Lösungen, Praktische Übungen am Rechner / Labor, Moodle
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> <p>NXP, UserManual LPC178x/7x User manual, UM10470, Rev. 4.0 — 21 December 2016</p>

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	4 oder 6
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, • 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	SAP-Systeme – Schnittstellen und ABAP-Programmierung
Kürzel	SAPrg
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	6
Angebotsturnus	jährlich
Dauer des Moduls	einsemestrig
Modulverantwortliche(r)	Prof. Dr. Jürgen Terpin
Dozent(in)	Dipl.-Ing. (FH) Karl Esau
Sprache	Deutsch
Zuordnung zum Curriculum	Wahlpflichtmodul
Nutzung in anderen Studiengängen	Bachelor Betriebswirtschaft, Bachelor Visual Computing
Zulassungsvoraussetzungen	Vorrückensberechtigung nach SPO
Inhaltliche Voraussetzungen	
Qualifikationsziele / Kompetenzen	Fachlich-methodische Kompetenzen: <ul style="list-style-type: none"> • Studierende sollen grundlegende Konzepte von Datenbanksystemen kennen und erklären können. • Studierende sollen Datenbanksysteme zielgerichtet verwenden können. • Studierende sollen Methoden und Techniken zum Entwurf von Datenbanken in Rahmen einer nicht-trivialen Anwendungsfragestellung verwenden können. • Studierende sollen Datenbankabfragen in Rahmen einer nicht-trivialen Anwendungsfragestellung zielgerichtet entwerfen können.

	<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"> • Studierende werden mit dem prinzipiellen Aufbau eines SAP-Systems, dessen Schnittstellen, der Abbildung technischer und betriebswirtschaftlicher Geschäftsprozesse, der Analyse des hierfür zugrundeliegenden Datenmodells und der Laufzeitumgebung vertraut gemacht. • Das Data-Dictionary und die Programmiersprachen ABAP und ABAP OO werden mit Syntax und Semantik vorgestellt und in zahlreichen Übungen vertieft. <p>Wichtige Aspekte der Software-Entwicklung auf einem SAP-System für die Entwicklung von User Interfaces wie Versionierung, Transport von Objekten, Debugging werden vermittelt.</p>
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 Minuten), praktischer Leistungsnachweis (Gewicht 2:1)
Sonstige Leistungsnachweise	
Medienformen	Beamer, Tafel, E-Learning Medien, SAP-GUI am PC/Notebook
Literatur	<ul style="list-style-type: none"> • Vorlesung/Skript in Buchform • Schwaiger, R. (2019): Schrödinger programmiert ABAP – Das etwas andere Fachbuch, 3. Aufl., Rheinwerk Verlag, Bonn. • Franz, T.; Trapp, T. (2014): Anwendungsentwicklung mit ABAP Objects, 2. Aufl., Galileo Press, Bonn; Boston, Mass.

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, Informatik
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	Sicherheitskonzepte & Frameworks Sichere Architekturen & Entwicklung Sichere Programmierung mit Rust Test von Sicherheitsfunktionalität
Endnotenbildende Studien- / Prüfungsleistungen	Studien- und Projektarbeit
Sonstige Leistungsnachweise	-
Medienformen	Beamer und Tafel/Whiteboard, Elektronische Skripten und Arbeitsunterlagen PC-Laborprojekt mit virtualisierter Übungsumgebung
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	4 oder 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"> ○ Automatisierung von Kommandofolgen ○ Variablen ○ Fallunterscheidungen, Schleifen ○ Verarbeitung von Zeichenketten • Prozesse in Linux <ul style="list-style-type: none"> ○ Erzeugung und Beendigung ○ Wertübergabe und Wertrückgabe ○ Signale zum Steuern von Prozessen ○ Vordergrund- und Hintergrundprozesse • Interprozesskommunikation in Linux <ul style="list-style-type: none"> ○ Umlenkung von Kanälen/Pipes (in der Shell) ○ Semaphoren (im C-Programm, System-API) ○ Shared Memory (im C-Programm, System-API) • Entwicklungswerkzeuge <ul style="list-style-type: none"> ○ Generierung ausführbarer Code (C-Compiler) ○ Fehlersuche
Endnotenbildende Studien- / Prüfungsleistungen	Schriftliche Prüfung (90 min) und praktische Leistungsnachweise im Rahmen einer praktische Studienarbeit im Verhältnis 1:1
Sonstige Leistungsnachweise	-
Medienformen	Beamer, Tafel; Lernmanagementsysteme (Moodle), Elektronisches Skript und elektronische Arbeitsunterlagen; PC-Systeme;
Literatur	Antonova R, Slaveva V., Slavova T.; Grundlagen und Praxis der Bash- und C-Programmierung in Unix/Linux, Pearson Deutschland GmbH, München, 2022. Cooper M: Advanced Bash-Scripting Guide; 2014; http://tldp.org/LDP/abs/html/index.html (abgerufen 26.10.2022) Plöttner J., Wendzel S.: Linux, Der Grundkurs; Rheinwerk, Bonn, 2021. Wolf J., Kania S.: Linux-UNIX-Programmierung; 4. Auflage, Rheinwerk, Bonn, 2016. Wolf J., Kania S., Sommer F.: Shell-Programmierung; 7.

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	4 oder 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	Fachlich-methodische Kompetenzen: Studierende sollen <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

	<ul style="list-style-type: none"> • die Rolle von Geschäftsprozessen als Anforderungsquelle verstehen, • 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	Rupp, C.: Requirements-Engineering und -Management. Hanser, 6. Auflage, 2015 Wieggers K., Beatty, J.: Software Requirements. Microsoft Press, Redmond, 3. Auflage, 2013 Drescher, A., Koschmider, A., Oberweis, A.: Modellierung und Analyse von Geschäftsprozessen. De Gruyter Oldenbourg, 2017 Gadatsch, A.: Grundkurs Geschäftsprozess-Management. Vieweg-Teubner, 7. Auflage, 2012 McConnell, S.: Software Estimation. Microsoft Press, Redmond, 2006

	Weiterführende Spezialliteratur für die verschiedenen Kapitel
--	---------------------------------------------------------------

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	4 oder 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	Fachlich-methodische Kompetenzen: Studierende sollen <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

	<p>Artefakte (z.B. Anforderungen, Code, Modelle) kennen, verstehen und anwenden können.</p> <p>Persönliche-Soziale Kompetenzen: 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	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	7
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 • Design

	<ul style="list-style-type: none"> ○ Entwurfskriterien ○ Standard-Softwarearchitekturen ● Softwaretest <ul style="list-style-type: none"> ○ Testprozess ○ 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	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, 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 Pxpark Erlangen, 2016, ISBN 978-3-89578-455-2.</p> <p>PLCopen: www.plcopen.org</p> <p>sowie weitere Bücher und URL Links</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	6
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.

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 Threadsynchronisation • 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	<p>Abts D.: Masterkurs Client/Server-Programmierung mit Java, 4. Auflage; Vieweg, 2015.</p> <p>Bengel G.: Verteilte Systeme, 4. Auflage; Springer Vieweg, Wiesbaden, 2014.</p> <p>Coulouris G., Dollimore J., Kindberg T., Blair G.: Distributed Systems, Concepts and Design; 5. Auflage; Pearson, 2012.</p> <p>Tanenbaum A., van Steen M.: Distributed Systems, Principles and Paradigms; 2. Auflage, Prentice Hall Pearson, 2014.</p>

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, • ihren Standpunkt mündlich zu vertreten, andere Meinungen in der Diskussion zu respektieren und die

	<p>daraus gewonnenen Erkenntnisse zu reflektieren,</p> <ul style="list-style-type: none"> • 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