

Modulhandbuch

Bachelor Studiengang

Informatik

FPO Februar 2025

Version: Wintersemester 2025/26

Stand: 01.09.2025

Studienverlaufsplan

1.Semester	2.Semester	3.Semester	4.Semester	5.Semester	6.Semester
Wissenschaftliches Arbeiten	Programmierpraktikum (Einführung in das projektorientierte Arbeiten)	Grundlagen der Betriebssysteme	Webentwicklung Frontend	Webentwicklung Backend	Wahlpflichtmodul (9/9)
Grundlagen der Informatik 1	Grundlagen der Informatik 2	Grundlagen der Informatik 3	IT-Projektmanagement	Softwareengineering	Projektarbeit
Mathematik 1	Mathematik 2	Rechnernetze	Wahlpflichtmodul (3/9)	Wahlpflichtmodul (6/9)	Bachelorarbeit
Einführung in die Programmierung	Objektorientierte Programmierung	Wahlpflichtmodul (1/9)	Wahlpflichtmodul (4/9)	Wahlpflichtmodul (7/9)	
Rechnerarchitektur	Datenbanken 1	Wahlpflichtmodul (2/9)	Wahlpflichtmodul (5/9)	Wahlpflichtmodul (8/9)	Kolloquium

Inhaltsverzeichnis

Studienverlaufsplan	2
Pflichtmodule des 1. Semesters	5
Wissenschaftliches Arbeiten	5
Grundlagen der Informatik 1	7
Mathematik 1	9
Einführung in die Programmierung	11
Rechnerarchitektur	13
Pflichtmodule des 2. Semesters	15
Programmierpraktikum (Einführung in das projektorientierte Arbeiten)	15
Grundlagen der Informatik 2	17
Mathematik 2	19
Objektorientierte Programmierung	21
Datenbanken 1	23
Pflichtmodule des 3. Semesters	25
Grundlagen der Betriebssysteme	25
Grundlagen der Informatik 3	27
Rechnernetze	29
Pflichtmodule des 4. Semesters	31
Webentwicklung Frontend	31
IT-Projektmanagement	33
Pflichtmodule des 5. Semesters	35
Webentwicklung Backend	35
Softwareengineering	37
Pflichtmodule des 6. Semesters	39
Projektarbeit	39
Bachelorarbeit	41
Kolloquium	42
Wahlpflichtmodule	43
Betriebswirtschaftslehre	43
Cloud Computing	45
Datenbanken 2	47
Datenschutz	49
Deep Learning	50
Effiziente Algorithmen	52
Einführung in die Theoretische Informatik	54
Einführung Machine Learning	56
Fortgeschrittene Internettechnologien	58
Funktionale Programmierung	60
Frontend-Frameworks für Webanwendungen	61
Gender und Diversity in der Informatik	63
Hardwarenahe Programmierung	65
IT-Recht	67
Marketing	68
Mobile Applikationen	70
Natural Language Processing	72
Operations Research	74
Ökosysteme	75
Partizipatives Design	77
Praktische Anwendungen von Algorithmen	79
Praktische Betriebssysteme 1	81
Praktische Betriebssysteme 2	83
Programmierung graphischer Benutzeroberflächen mit Java	84
Quantencomputing	85
Rechnernetze 2	87
Rechnungswesen 1	89
Rechnungswesen 2	91

Statistik.....	93
Systembiologie.....	95
Umweltinformationssysteme (UIS).....	97
Virtualisierung	99
Container	101
Container Anwendungsentwicklung	102
Cloud-native Software Developement	102
Container Künstliche Intelligenz	104
Container Systemintegration	105
Container Querschnittsthemen	106

Pflichtmodule des 1. Semesters

Wissenschaftliches Arbeiten

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 1. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlangen praktische Recherchefähigkeiten. Sie können Quellen auswählen und beurteilen und kennen grundlegende Methoden und Kriterien wissenschaftlichen Arbeitens in der Informatik. Die Studierenden kennen unterschiedliche Textarten und können selbständig einfache Texte mit methodisch-wissenschaftlichem Anspruch schreiben. Zu den hier zu erwerbenden Kompetenzen zählen die selbständige Literaturrecherche, korrektes Zitieren, Erstellen eines Exposé, zielgruppengerechtes Schreiben sowie der Umgang mit dem Textsatzsystem LaTeX.				
3	Inhalte <ul style="list-style-type: none">• Prinzipien wissenschaftlichen Arbeitens, Kriterien für wissenschaftliche Qualität• Vorgehensweise und Zeitplanung beim wissenschaftlichen Arbeiten• Techniken der Bibliotheksrecherche & Quellenbeschaffung, -bewertung, -auswertung, mit Anwendung auf IT-Fragestellungen, Angabe von Fundstellen / Quellen, Zitat / Plagiat• Erstellen und Abstimmen eines Exposés• Textarten, Kriterien zu sprachlicher Qualität von Texten, Schreibprozess / Bearbeitung von Texten• Präsentation, Poster, Bericht / schriftliche Ausarbeitung• Erstellen und Gestalten von Abbildungen und Tabellen• Grundzüge Urheberrecht, Open Access / Open Data• KI & wissenschaftliches Arbeiten: Einsatzmöglichkeiten und Grenzen, verantwortlicher Umgang• Umgang mit Bias				
4	Lehr- und Lernformen Vorlesung / seminaristischer Unterricht (50%), Anwendung (50%) mit Rechercheübungen in Zusammenarbeit mit der Bibliothek, praktische Übungen (Textsatz / Formelsatz, Quell-/ Literaturdatenbank), Textwerkstatt, Arbeit in Kleingruppen, Einübung von Präsentation wissenschaftlicher Erkenntnisse				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Prozessorientierte Prüfungsleistung (gemäß FPO) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Balzert et al, Wissenschaftliches Arbeiten: Ethik, Inhalt & Form wiss. Arbeiten, Handwerkszeug, Quellen, Projektmanagement, Präsentation, Springer Campus, 2017. Bänsch et al., Wissenschaftliches Arbeiten, De Gruyter Oldenbourg, 2020.

Grundlagen der Informatik 1

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 1. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2SWS / 22,5 h Vorlesung b) 3 SWS / 33,75 h praktische Übung		Kontaktzeit 5 SWS / 56,25 h	Selbststudium 123,75 h	geplante Gruppengröße b) bis 20 Studierende
2	Lernergebnisse (learning outcomes) <ul style="list-style-type: none"> • fundamentale Paradigmen, Fakten und Methoden der angewandten Informatik kennen und anwenden können • exemplarische Algorithmen für grundlegende Probleme der Informatik kennen und verstehen (algorithmisches Denken) • erste, einfache Methoden für den Entwurf und die Analyse effizienter Algorithmen kennen und anwenden können (Rekursion, Memoisation) • einfache Aufwandsschätzungen für IT-Systeme durchführen können Kompetenzen: <ul style="list-style-type: none"> • algorithmisches Denken • Methodenkompetenz • Besprechungen mit Mitstudierenden unter Anleitung planen, initiieren und durchführen können • Mitarbeit in und Steuerung von Gruppenprozessen (Moderation beim Lernteamcoaching) 				
3	Inhalte <ul style="list-style-type: none"> • Informationsdarstellung, Zahldarstellung: Ganz-/Festpunktzahlen, inkl. arithmetischer Operationen • Gleitpunktzahlen nach IEEE 754, inkl. arithmetischer Operationen, Genauigkeit der Darstellung • Einführung in algorithmisches Denken am Beispiel vom "Schnellen Potenzieren" und einfachen mathematischen / zahlentheoretischen Algorithmen • Darstellung von Geoinformation, einfache Geoalgorithmen • Aufwandsschätzung • Laufzeitmessung auf Basis eigener Implementierungen: Messmethodik, Ergebnisdarstellung • einfache Optimierungsprobleme • Einführung in Graphen • reguläre Ausdrücke, Sprachen, Funktionen und Quantoren 				
4	Lehr- und Lernformen <p>Studierenden-zentriertes Lernen mit Arbeit in Kleingruppen (5-6 Studierende: Lernteamcoaching und Gruppenpuzzle), im Plenum sowie in Einzelarbeit.</p> <p>Unter Anleitung der Lehrenden planen und steuern die Studierenden ihren eigenen Lernprozess und können dazu auf ausgewählte Texten aus Lehrbüchern, die mittels Lernteamcoaching gemeinsam erarbeitet werden, sowie auf einen Pool von vorbereiteten strukturierten Lernmaterialien zu den jeweiligen Einzelthemen zurückgreifen. Dazu Aufgaben auf Moodle, Stationenlernen, Lernkartei sowie nachbereitende Vorlesung zur Lernstandssicherung. Der Lernfortschritt wird durch veranstaltungsbegleitende Teilprüfungen gesichert.</p> <p>In Zusammenarbeit mit dem parallel stattfindenden Modul "Grundlagen der Informatik 3" wird dabei auch semesterübergreifend gearbeitet. Die fortgeschrittenen Studierenden leiten dabei die Erstsemester an, geben ihnen individuell Wissen zum Stoff von "Grundlagen der Informatik 1" weiter, stehen für Rückfragen/ Fachgespräche dazu zur Verfügung und begleiten sie bei ihrem Lernprozess.</p> <p>Die drei Module "Grundlagen der Informatik 1", "Grundlagen der Informatik 2" und "Grundlagen der Informatik 3" bilden sowohl inhaltlich als auch vom Kompetenzerwerb her einen zusammenhängenden Zyklus: Im Modul "Grundlagen der Informatik 1" wird die Basis für das Folgemodul "Grundlagen der Informatik 2" gelegt. Zugleich unterstützen die erfahreneren Studierenden aus "Grundlagen der Informatik 3" den Einstieg ins selbständige Lernen.</p>				

5	Teilnahmevoraussetzungen Formal: Inhaltlich:
6	Prüfungsformen Prozessorientierte Prüfungsleistung (gemäß FPO) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)
7	Prüfungsvorleistung keine
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Cormen, Leiserson, Rivest, Stein, Algorithmen - Eine Einführung, Oldenburg Wissenschaftsverlag Gumm, Sommer, Informatik, De Gruyter de Lange, Geoinformatik, Springer (Kap. 2, 4.1., 4.2) Kopec, Algorithmen in Python, Rheinwerk Computing

Mathematik 1

Kennnummer		Workload 180 h	Credits 6 CP	Studiensemester 1. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 4 SWS / 45 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 6 SWS / 67,5 h		Selbststudium 112,5 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Dieses Modul legt die Grundlagen der Differenzial- und Integralrechnung für Funktionen einer Veränderlichen. Die Studierenden sollen dabei das Verständnis für die grundlegenden Prinzipien der Analysis entwickeln und die Grundbegriffe und -techniken sicher beherrschen können. Darüber hinaus sollen die mathematische Arbeitsweise an konkreten Fragestellungen erlernt, mathematische Intuition entwickelt und die Entwicklung der Analysis exemplarisch an zentralen Begriffen nachvollzogen werden.					
3	Inhalte <ul style="list-style-type: none">• Mengen und Mengenoperationen, Zahlensysteme• Gleichungen und Ungleichungen sowie Binomischer Lehrsatz• Folgen• Funktionen und Kurven• Differenzialrechnung• Integralrechnung					
4	Lehr- und Lernformen Vorlesungen mit begleitenden Übungen in Kleingruppen (< 25 Teilnehmer*innen) Die vorgestellten Verfahren werden an Beispielen angewandt und vertieft. Die Ergebnisse werden in Form von Impulsvorträgen vorgestellt und im Plenum diskutiert.					
5	Teilnahmevoraussetzungen Formal: Inhaltlich:					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Übung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam, Uwe Gogolin					

12	Sonstige Informationen Literaturauswahl: Papula, Lothar; Mathematik für Ingenieure und Naturwissenschaftler Band 1; Vieweg + Teubner, Wiesbaden Papula, Lothar; Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler; Vieweg + Teubner, Wiesbaden Papula, Lothar; Mathematik für Ingenieure und Naturwissenschaftler Anwendungsbeispiele; Vieweg, Wiesbaden
----	--

Einführung in die Programmierung

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 1. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung c) 2 SWS / 22,5 h Praktikum	Kontaktzeit 6 SWS / 67,5 h	Selbststudium 112,5 h	geplante Gruppengröße c) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Python gilt als eine der zugänglichsten Programmiersprachen für Anfänger*innen und genießt eine breite Akzeptanz in der Industrie, insbesondere durch seine Anwendung in vielen Bereichen der Informatik. In diesem Kurs erhalten die Studierenden einen umfassenden Einblick in die Grundlagen der Programmierung mit Python. Sie lernen, wie sie die Sprache effektiv für die Lösung realer Probleme einsetzen können. Durch praktische Beispiele und Übungen entwickeln sie ein intuitives Verständnis für den Aufbau und die Funktionsweise von einfachen Programmen. Darüber hinaus zeigt der Kurs zeigen, Python als eine Multi-Paradigmen Sprache die Prinzipien des prozeduralen, objektorientierten und in begrenztem Umfang auch des funktionalen Programmierens unterstützt. Die Kursstruktur fördert das kontinuierliche Lernen und die Anwendung der theoretischen Konzepte durch regelmäßige praktische Übungen und Projekte. Am Ende des Kurses werden die Studierenden in der Lage sein, eigene kleine bis mittelgroße Python-Programme zu schreiben, bestehenden Code zu analysieren und zu verbessern und grundlegende Algorithmen und Datenstrukturen zu implementieren. Die Ziele des Kurses sind: <ul style="list-style-type: none">• die grundlegenden Funktionen der Programmiersprache Python zu vermitteln• den Studierenden das Schreiben von prozeduralen, sowie grundlegend auch objektorientierten und funktionalen Programmen in Python zu ermöglichen• wichtige Datenstrukturen und Algorithmen anhand von Python Programmen einzuführen• den Funktionsumfang von Programmen durch Verwendung geeignete Module aus der Python-Standardbibliothek zu erweitern• die Studierenden zu dabei zu unterstützen, ein gutes Verständnis für die Grundlagen der Programmierung und von Entwurfsmustern zu erlangen• Professionelle Programmiertechniken und einen guten Programmierstil zu vermitteln Nach Abschluss des Kurses werden die Studierenden in der Lage sein: <ol style="list-style-type: none">1. Grundlegende Python-Programme zu entwickeln und zu debuggen.2. Die Konzepte der prozeduralen und objektorientierten Programmierung zu verstehen und anzuwenden.3. Verschiedene Datenstrukturen in Python effektiv zu nutzen.4. Modularität und wiederverwendbaren Code durch das Verständnis von Funktionen und Klassen zu fördern.5. Grundlegende Algorithmen zu implementieren und deren Anwendung zu verstehen.				
3	Inhalte Einführung in die (Python-) Programmierung <ul style="list-style-type: none">• Ausführen von Python-Code (interaktive Sitzungen, Befehlszeile, Jupyter Notebooks)• Grundlegende Sprachelemente (Datentypen und Operationen, Bezeichner, Skalare- und Sequenzielle-Datentypen, Kontrollfluss, String-Manipulation) Funktionen <ul style="list-style-type: none">• Argumente• Rückgabewerte• Rekursion...				

	<p>....</p> <p>Ein-/Ausgabe</p> <ul style="list-style-type: none"> • Standardausgabe • Dateien <p>Modularisierung</p> <ul style="list-style-type: none"> • Module und Pakete • Standardbibliothek <p>Nicht-imperative Programmierparadigmen</p> <ul style="list-style-type: none"> • Grundlagen der Objektorientierung (Klassen und Objekte, Attribute und Methoden, Vererbung) • Funktionale Programmierung (Lambda Funktionen, Funktionen höherer Ordnung) • Aspektorientiert Programmierung (Dekoratoren) <p>Programmierstil</p> <ul style="list-style-type: none"> • Ausnahmebehandlung • Testen von Programmen • Effiziente Programmierung, Laufzeiten messen/Benchmarking <p>Ausgewählte Pakete der Standardbibliothek</p> <ul style="list-style-type: none"> • Systemzugriffe: Interaktion mit dem Betriebssystem und der Laufzeitumgebung, Ausführung von Systemprozessen (z.B. os, sys, subprocess) • Mathematik und Statistik: Grundlegende mathematische Operationen und Zufallszahlen (z.B. math, random) • Daten und I/O: Lesen und Schreiben von JSON und CSV-Dateien, Objektserialisierung (z.B. json, csv, pickle, io) • Webzugriffe und Web-Services: Erstellung von HTTP-Services und Arbeiten mit URLs, oder Netzwerkschnittstellen (z.B. http, urllib, socket)
4	<p>Lehr- und Lernformen</p> <p>Vorlesung mit begleitenden Übungen und Praktika (in Kleingruppen)</p>
5	<p>Teilnahmevoraussetzungen</p> <p>Formal:</p> <p>Inhaltlich:</p>
6	<p>Prüfungsformen</p> <p>Klausur</p> <p>(Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)</p>
7	<p>Prüfungsvorleistung</p> <p>keine</p>
8	<p>Voraussetzungen für die Vergabe von Kreditpunkten</p> <p>Bestehen der Modulprüfung</p>
9	<p>Verwendung des Moduls (in anderen Studiengängen)</p>
10	<p>Stellenwert der Note für die Endnote</p> <p>$6/180 = 3,33\%$</p>
11	<p>Modulbeauftragte*r und hauptamtlich Lehrende</p> <p>Prof. Dr. Heiner Giefers</p>
12	<p>Sonstige Informationen</p> <p>Literaturauswahl:</p> <p>Charles R. Severance: Python for Everybody: Exploring Data in Python 3</p> <p>Eric Matthes: <i>Python Crashkurs</i>, dpunkt Verlag</p> <p>Wes McKinney, <i>Datenanalyse mit Python</i>, O'Reilly</p> <p>Johannes Ernesti, Peter Kaiser, <i>Python 3 – Das umfassende Handbuch</i>, Rheinwerk</p>

Rechnerarchitektur

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 1. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4SWS / 45h	Selbststudium 135h	geplante Gruppengröße b) 20 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erhalten einen Einblick in die Technologie und Architektur moderner Computersysteme. Sie können den Aufbau eines Rechners beschreiben und das Zusammenwirken von Hardware und Software erklären. Sie kennen die Grundprinzipien von Befehlssätzen und können kleine Assemblerprogramme für einen bekannten Befehlssatz entwickeln. Die Studierenden können Elemente der Prozessorarchitektur auf Mikroarchitekturebene benennen und Methoden zur Leistungsbewertung von Prozessoren einsetzen.				
3	Inhalte <ul style="list-style-type: none"> • Aufbau von Computersystemen • Grundlagen der Digitaltechnik • Befehlssätze • Mikroarchitektur • Speicherhierarchie • Leistungsbewertung • Moderne Prozessorarchitekturen 				
4	Lehr- und Lernformen Vorlesung und Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$				

11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: David A. Patterson und John L. Hennessy: <i>Rechnerorganisation und Rechnerentwurf – Die Hardware/Software-Schnittstelle</i> , De Gruyter, 2016 David A. Patterson und John L. Hennessy: <i>Computer Organization and Design – The Hardware/Software Interface. MIPS Edition</i> , Morgan Kaufmann, 2013 John P. Hayes: <i>Computer Architecture and Organization</i> , McGraw-Hill, 1998 John P. Hayes: <i>Introduction to Digital Logic Design</i> , Addison-Wesley, 1993 Axel Böttcher: <i>Rechneraufbau und Rechnerarchitektur</i> . Springer, 2006

Pflichtmodule des 2. Semesters

Programmierpraktikum (Einführung in das projektorientierte Arbeiten)

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 2. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) bis 20 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen gängige Werkzeuge der Softwareentwicklung und Versionsverwaltung und können diese für die eigene Programmierfähigkeit und für die projektorientierte Zusammenarbeit in Kleingruppen produktiv einsetzen. Die Studierenden kennen Designprinzipien für Clean Code und wenden diese praktisch an. Für vorgegebene Programmieraufgaben mit moderater Komplexität können sie die Zerlegung in Teilaufgaben vornehmen und die zeitliche und organisatorische Zusammenarbeit in der Kleingruppe selbständig planen sowie mit den Lehrenden abstimmen. Sie können den Fortschritt der Arbeit ihrer Kleingruppe selbständig dokumentieren und berichten / vorstellen; dafür kennen und nutzen sie gängige Methoden und Tools. Die Studierenden können Programme automatisiert testen und aussagekräftige Fehlerberichte erstellen. Sie kennen die Grundlagen zu Continuous Integration / Continuous Delivery sowie grundlegende Werkzeuge dazu.				
3	Inhalte <ul style="list-style-type: none">• Clean Code / Designprinzipien für effizient wartbaren Programmcode, Vermeidung typischer Fallstricke• Versionsverwaltung & Softwareentwicklungsumgebung, Einführung in die Nutzung von Werkzeugen für Entwicklung & Versionsverwaltung• Arbeit an vorgegebenen, strukturierten Programmierprojekten in der Kleingruppe• systematische Fehlersuche, Unittests• Tools für Profiling• Dokumentation / Reporting, inkl. Tools• Einführung in die Prozesse und Tools für Continuous Integration / Continuous Delivery				
4	Lehr- und Lernformen Vorlesung und problemorientiertes Lernen in Kleingruppen zu je 4-5 Studierenden				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Programmierkenntnisse aus dem 1. Semester				
6	Prüfungsformen Prozessorientierte Prüfungsleistung (§ 15 FPO) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Sweigart, Best Practices for Writing Clean Code / Der Weg zum Python-Profi: ein Best-Practice-Buch für sauberes Programmieren, dpunkt.verlag, 2022. Scott Chacon, Ben Straub, Pro Git, Apress Publishing, 2020. Robert C. Martin, Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code, MIT press. Anaya, Mariano. Clean Code in Python: Develop maintainable and efficient code, Packt Publishing, 2021. Lenz, Moritz. Python Continuous Integration and Delivery: A Concise Guide with Examples, Apress, 2018.

Grundlagen der Informatik 2

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 2. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen 3 SWS / 33,75 h praktische Übung		Kontaktzeit 3 SWS / 33,75 h	Selbststudium 146,25 h	geplante Gruppengröße bis 20 Studierende
2	Lernergebnisse (learning outcomes) <ul style="list-style-type: none"> • grundlegende Datenstrukturen (Listen, Bäume, Graphen) mit ihren Eigenschaften und Anwendungen kennen und einsetzen können • weitere exemplarische Algorithmen für grundlegende Probleme der Informatik kennen und verstehen (Baumdurchlauf, einfache Graphalgorithmen, Suchen und Sortieren) • elementare Methoden für die Analyse von Algorithmen kennen und selbst anwenden können • einfache reguläre Ausdrücke verstehen, selbst erzeugen und für Patternmatching anwenden können • Besprechungen untereinander und mit den Lehrpersonen eigenständig planen, initiieren und durchführen können Kompetenzen: <ul style="list-style-type: none"> • algorithmisches Denken • Methodenkompetenz • Analysefähigkeit • Synthesefähigkeit • den eigenen Lernprozess effizient steuern und reflektieren können • Mitarbeit in und Steuerung von Gruppenprozessen 				
3	Inhalte <ul style="list-style-type: none"> • Bäume & Graphen, inkl. Graphdarstellung / Graphvisualisierung in Python • Eigenschaften von Graphen (Zusammenhang, Eulerkreisproblem), einfache Algorithmen zum Graphdurchlauf (Breiten-/Tiefendurchlauf) • effiziente Such- und Sortierverfahren in Feldern • Einführung in reguläre Ausdrücke, Patternmatching • einfache Analyse von Algorithmen (Terminierung, numerische Stabilität) • Einführung in O-Notation, Laufzeitklassen, einfache Abschätzungen 				
4	Lehr- und Lernformen Problembasiertes Lernen, kooperatives Lernen / Lernen durch Lehren, Lernteamcoaching. Die Studierenden arbeiten sich in Kleingruppen per angeleitetem Lernteamcoaching auf Basis vorgegebener Texte selbstständig in Themen der Veranstaltung ein. Sie bereiten dann angeleitet die Wissensweitergabe an Peers vor. Dazu strukturierten sie das erarbeitete Wissen in mehrere einzelne Lerneinheiten und bereiten unterstützende Abbildungen, Programmbeispiele und auch Übungsaufgaben zur Darstellung ihres erworbenen Wissens vor. Die Zuhörenden erhalten dadurch leichteren Zugang zu den Lehrtexten. Beim Nacharbeiten der Lehrtexte können die erstellten Programmbeispiele genutzt werden und die Peers stehen für Rückfragen bereit.				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: "Grundlagen der Informatik 1", Programmierkenntnisse aus dem 1. Semester				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Übung: Bearbeitung von Aufgaben in Moodle – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				

8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Beutelspacher, Zschiegner, Diskrete Mathematik für Einsteiger: Mit Anwendungen in Technik und Informatik, Vieweg und Teubner Cormen, Leiserson, Rivest, Stein, Algorithmen -Eine Einführung, Oldenbourg Wissenschaftsverlag Al-Taie, Python for Graph and Network Analysis, Springer Nature Stubblebine, Reguläre Ausdrücke: kurz & gut, O'Reilly

Mathematik 2

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 2. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Dieses Modul legt die Grundlagen der Linearen Algebra und gibt eine Einführung in die Kombinatorik und die Grundlagen der Statistik. Die Studierenden sollen dabei das Verständnis für die grundlegenden Prinzipien der Linearen Algebra, Wahrscheinlichkeitsrechnung und Statistik entwickeln und die Grundbegriffe und -techniken sicher beherrschen können. Darüber hinaus sollen die mathematische Arbeitsweise an konkreten Fragestellungen erlernt, mathematische Intuition entwickelt und die Entwicklung der Linearen Algebra, Wahrscheinlichkeitsrechnung und Statistik exemplarisch an zentralen Begriffen nachvollzogen werden.</p>				
3	Inhalte <ul style="list-style-type: none"> • Vektoren, Verknüpfungen von Vektoren • Vektorrechnung im 2- und 3-dimensionalen Raum • Matrizen und Determinanten • Lösung linearer Gleichungssysteme • Eigenwerte und Eigenvektoren • Wahrscheinlichkeitsrechnung • Grundlagen der Statistik 				
4	Lehr- und Lernformen <p>Vorlesungen mit begleitenden Übungen in Kleingruppen (< 25 Teilnehmer*innen)</p> <p>Die vorgestellten Verfahren werden an Beispielen angewandt und vertieft. Die Ergebnisse werden in Form von Impulsvorträgen vorgestellt und im Plenum diskutiert.</p>				
5	Teilnahmevoraussetzungen <p>Formal:</p> <p>Inhaltlich:</p>				
6	Prüfungsformen <p>Klausur</p> <p>(Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)</p>				
7	Prüfungsvorleistung <p>Studienleistung für Übung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert</p>				
8	Voraussetzungen für die Vergabe von Kreditpunkten <p>Bestehen der Modulprüfung</p>				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33 %
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam, Uwe Gogolin
12	Sonstige Informationen Literaturauswahl: Papula, Lothar: Mathematik für Ingenieure und Naturwissenschaftler Band 1; Vieweg + Teubner, Wiesbaden Papula, Lothar: Mathematik für Ingenieure und Naturwissenschaftler Band 2; Vieweg + Teubner, Wiesbaden Papula, Lothar: Mathematik für Ingenieure und Naturwissenschaftler Band 3; Vieweg + Teubner, Wiesbaden Papula, Lothar: Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler; Vieweg + Teubner, Wiesbaden Papula, Lothar: Mathematik für Ingenieure und Naturwissenschaftler Anwendungsbeispiele; Vieweg, Wiesbaden

Objektorientierte Programmierung

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		180 h	6 CP	2. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Vorlesung vermittelt grundlegende Kenntnisse der Programmiersprache Java und stellt in Auszügen die umfangreichen Bibliotheken der Java Standard Edition vor.					
3	Inhalte <ul style="list-style-type: none">• Grundlegende Syntax von Java• Objektorientierung in Java• Ausgewählte Bibliotheken der Java-Plattform• Im Praktikum werden auf die Vorlesung abgestimmte Präsenzaufgaben bearbeitet.					
4	Lehr- und Lernformen Vorlesung mit Praktikum					
5	Teilnahmevoraussetzungen Formal: Inhaltlich:					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Praktikum– die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					

11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Andreas Steins
12	Sonstige Informationen Literaturauswahl: Ken Arnold, James Gosling, Java, Die Programmiersprache, Addison-Wesley Helmut Balzert, Java 5: Objektorientiert programmieren, W3L-Verlag Bruce Eckel, Thinking in Java Web: " http://www.BruceEckel.com ", als Buch bei Prentice-Hall Friederich Esser, Java 2, Web: " http://download.galileo-press.de/openbook/java2/galileocomputing_java2.zip ", als Buch bei Galileo Press David Flanagan, Java in a Nutshell, O'Reilly Erich Gamma, Ralph Helm, Richard Johnson, John Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley Guido Krüger, Handbuch der Java-Programmierung, Web: " http://www.javabuch.de/ ", als Buch bei Addison-Wesley

Datenbanken 1

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 2. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Modul Datenbanken 1 soll vertiefte Kenntnisse in der Datenmodellierung, der Nutzung der Structured Query Language (SQL) vorwiegend auf eine Tabelle und dem Zusammenwirken mit Programmiersprachen vermitteln.				
3	Inhalte <ul style="list-style-type: none">• Einführung in die Datenmodellierung• Normalisierung• Prinzipieller Aufbau einer relationalen Datenbank• kurze Einführung ins Relationenmodell• Vorgehensmodell zur Herleitung einer Datenbankstruktur• Einführung in SQL, insbes. Tabellenstrukturanweisungen, Abfragen einzelner Tabellen, einfache Verbundanweisungen, Anzeigeaufbereitungen• Einbindung von SQL in Programmierungsumgebungen über standardisierte Schnittstellen				
4	Lehr- und Lernformen Vorlesungen, seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: „Grundlagen der Informatik 1“, „Einführung in die Programmierung“ Inhaltlich:				
6	Prüfungsformen Klausur und Kombination mit Vorleistung aus dem Praktikum/seminaristischem Unterricht oder prozessorientierte Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				
10	Stellenwert der Note für die Endnote 6/180 = 3,33%				

11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Uwe Klug
12	Sonstige Informationen Literaturauswahl: R. Elmasri, S. B. Navathe, Grundlagen von Datenbanksystemen, Addison-Wesley Heuer, G. Saake, Datenbanken: Konzepte und Sprachen, Internat. Thomson Publ. M. Klettke, H. Meyer; XML & Datenbanken; dpunkt.verlag U. Klug; SQL - Der Einstieg in die deklarative Programmierung; W3L Verlag U. Klug; Datenbankanwendungen entwerfen & programmieren; W3L Verlag G.Lausen; Datenbanken; Spektrum Akademischer Verlag G.Saake, K.-U. Sattler; Datenbanken & Java; dpunkt.Verlag G.Lausen; Datenbanken; Spektrum Akademischer Verlag G.Saake, K.-U. Sattler; Datenbanken & Java; dpunkt.Verlag

Pflichtmodule des 3. Semesters

Grundlagen der Betriebssysteme

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 3. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Nach der Teilnahme an diesem Modul kennen die Studierenden die Kernfunktionen eines Betriebssystems und verstehen, wie Anwendungen durch das Betriebssystem von der Hardware „ferngehalten“ werden. Als wesentliche Mechanismen und Konzepte erkennen sie die Abstraktion und Virtualisierung, die stabiles Multitasking (also die scheinbar parallele Ausführung mehrerer Anwendungen) ermöglichen.</p> <p>Die Studierenden können kleine Anwendungen in C schreiben, mit denen sie Betriebssystem-Features wie parallele Programmausführung (mit mehreren Prozessen oder mehreren Threads) und Synchronisation testen.</p> <p>Durch ein vertieftes Verständnis der Abläufe im Betriebssystem achten die Studierenden auch bei der Software-Entwicklung darauf, keinen Code zu schreiben, der zu einer ungünstigen Nutzung der Rechnerressourcen führt.</p>				
3	Inhalte <ul style="list-style-type: none"> • Einführung in C und x86_64-Assembler • Prozesse und Threads • Geräte und Interrupts • Scheduling-Verfahren • Synchronisation und Deadlocks • Speicherverwaltung 				
4	Lehr- und Lernformen Vorlesung mit begleitenden Übungen (in Kleingruppen)				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen Literaturauswahl: Tanenbaum, Andrew S. und Bos, Herbert: <i>Moderne Betriebssysteme</i> , Pearson Studium Ehses, Erich; Köhler, Lutz; Riemer, Petra; Stenzel, Horst und Victor, Frank: <i>Systemprogrammierung in UNIX/Linux</i> , Vieweg+Teubner Stallings, William: <i>Operating Systems: Internals and Design Principles</i> Eßer, Hans-Georg und Freiling, Felix: <i>The Design and Implementation of the UNIX Operating System</i>

Grundlagen der Informatik 3

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 3. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 1 SWS / 11,25 h Vorlesung b) 3 SWS / 33,75 h praktische Übung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) bis 25 Studierende
2	Lernergebnisse (learning outcomes) <ul style="list-style-type: none"> fortgeschrittene Datenstrukturen kennen, einsetzen, und mit ihren Eigenschaften analysieren können Methoden für Entwurf und Analyse von Algorithmen kennen und anwenden können, exemplarische Algorithmen für grundlegende Probleme der Informatik kennen und verstehen (algorithmisches Denken) geeignete Datenstrukturen für typische Einsatzszenarien auswählen können, dabei das Zusammenspiel von Algorithmen und Computertechnik verstehen fachlich Feedback erarbeiten und geben können (am Beispiel von Lösungsansätzen für einfache Aufgaben zum Stoff des ersten Semesters) Besprechungen mit anderen Studierenden und mit Lehrpersonen eigenständig planen, initiieren und durchführen können Kompetenzen: <ul style="list-style-type: none"> algorithmisches Denken Methodenkompetenz Analysefähigkeit Synthesefähigkeit Geben von fachlichem Feedback 				
3	Inhalte <ul style="list-style-type: none"> fortgeschrittene Datenstrukturen balancierte Suchbäume, inkl. Laufzeitanalyse Hashing / Hashfunktionen Schneller Multiplikationsalgorithmus nach Karazuba Vertiefung zu Laufzeitanalyse und zur O-Notation Verarbeitung großer Datenmengen, Echtzeiten mit Praxisbezug (z.B. Geodaten, Klimadaten) 				
4	Lehr- und Lernformen Klassische Vorlesung und Übung werden ergänzt durch semesterübergreifendes „Lernen durch Lehren“, siehe Beschreibung beim parallel stattfindenden Modul "Grundlagen der Informatik 1". Veranstaltungsbegleitende Aufgaben auf Moodle, Prozessbegleitung und kontinuierliche Ergebnisdokumentation auf Moodle.				
5	Teilnahmevoraussetzungen Formal: „Grundlagen der Informatik 1“ Inhaltlich: "Grundlagen der Informatik 1", "Grundlagen der Informatik 2", "Mathematik 2" (Wahrscheinlichkeitsrechnung), Programmierkenntnisse aus den ersten beiden Semestern				
6	Prüfungsformen Prozessorientierte Prüfungsleistung (§ 15 FPO) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Cormen, Leiserson, Rivest, Stein, Algorithmen - Eine Einführung, Oldenbourg Wissenschaftsverlag Ottmann, Widmayer, Algorithmen und Datenstrukturen, Springer Solymosi, Grude, Grundkurs Algorithmen und Datenstrukturen in JAVA, Springer Vieweg

Rechnernetze

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 3. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Dieses Modul vermittelt die theoretischen und praktischen Kompetenzen zur bedarfsgerechten Planung und Weiterentwicklung sowie zum Betrieb der Netzwerkinfrastruktur eines Unternehmens. Hierbei werden insbesondere die praxisrelevanten Techniken und Protokolle zur Realisierung von Rechnernetzen betrachtet, wobei der Focus auf der Internet-Protokoll-Familie liegt. Aufbauend auf den vermittelten Grundlagen werden die Methoden und Protokolle zur Umsetzung der sicheren Kommunikation und der Übermittlung von multimedialen Inhalten über Rechnernetze behandelt. Zur Modellierung des Netzwerks werden das TCP/IP- und das OSI-Modell verwendet.				
3	Inhalte <ul style="list-style-type: none">• Aktive Komponenten, Strukturierte Verkabelung, Analysewerkzeuge• Ethernet: Standards, Topologie, Signalübertragung, Ethernetrahmen• Vermittlungsschicht: IPv4, IPv6, IP-Subnetting, Routing, ARP, RARP, ICMP• Transportschicht: UDP, TCP• Ausgewählte Protokolle und Dienste der Anwendungsschichten: Webserver (HTTP), Filetransfer (FTP), Email (SMTP, POP3, IMAP), Automatische Adressenvergabe (RARP, BootP, DHCP), Namensauflösung (Netbios, DNS, WINS)• Firewall: (Architekturen, DMZ, Paketfilter, Content Filter)• WLAN (Standards, Komponenten, Protokolle)• Multimediaanwendungen am Beispiel von VoIP sowie Audio- und Video-Streaming• Virtuelle Private Netze (L2TP, IPSec, SSL)• Netzwerkmanagement (SNMP, MIB)				
4	Lehr- und Lernformen Vorlesungen mit begleitenden Übungen in Kleingruppen (< 25 Teilnehmer*innen). Die in der Vorlesung vorgestellten Techniken und Protokolle werden im Rahmen des Praktikums in Testumgebungen praktisch aufgesetzt, konfiguriert und in Betrieb genommen. Die Ergebnisse werden protokolliert und in Kurzvorträgen vorgestellt und diskutiert.				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: „Mathematik 1“, „Mathematik 2“				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam
12	Sonstige Informationen Literaturauswahl: Kappes, Martin: Netzwerk- und Datensicherheit; Teubner Verlag Badach, Anatol: Hoffman, Erwin: Technik der IP-Netze; Hanser Washburn, Kevin: Evans, Jim: TCP/IP; Addison-Wesley Barth, Wolfgang: Das Firewall Buch, SuSE Press RFCs nach Ankündigung in der Vorlesung

Pflichtmodule des 4. Semesters

Webentwicklung Frontend

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180 h	6 CP	4. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können auf der Basis clientseitiger Technologien (HTML, CSS, JavaScript) einfache Webanwendungen erstellen und dabei existierende, einfache REST-APIs einbinden. Sie <ul style="list-style-type: none"> verstehen die Unterschiede zwischen einer Webanwendung und einer klassischen Anwendung und können die Vor- und Nachteile einer Webanwendung für einen konkreten Anwendungsfall beurteilen, kennen die Funktionsweise von Single-Page-Anwendungen und verstehen das dabei notwendige Zusammenwirken von Front- und Backend, und können Tools für das Testen und das Debugging von Frontendkomponenten verwenden. 				
3	Inhalte <ul style="list-style-type: none"> HTML Cascading Style Sheets JavaScript Authentisierungsverfahren im Web (OAuth, OpenID Connect, WebAuthn) Aktuelle Web-APIs (GeoLocation, LocalStorage, IndexedDB u.a.) Frameworks wie z.B. Bootstrap, React, Angular Anbindung von REST-APIs per HTTP 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl: Duckett, John, HTML & CSS: Erfolgreich Websites gestalten und programmieren, Weinheim P. Bühler, P. Schlaich, D. Sinner HTML5 und CSS3: Semantik - Design - Responsive Layouts, Berlin The Modern JavaScript Tutorial, online unter javascript.info

IT-Projektmanagement

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 4. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Seminar		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die grundlegenden Aufgaben, Vorgehensmodelle und Methoden des IT-Projektmanagements und können zugehörige Tools zur Projektplanung und Projektüberwachung einsetzen. Sie besitzen die theoretischen und praktischen Kompetenzen, um sich in einem Projekt zu orientieren, können zielorientiert im Projekt mitarbeiten und sind in der Lage, eigene Projekte selbständig zu planen, zu überwachen und zu steuern.				
3	Inhalte <ul style="list-style-type: none"> • Grundbegriffe: Projekt/ Projektarten, Projektziele, Projektphasen, Aufgaben des Pjm, Methoden, Instrumente und Werkzeuge des Pjm, Erfolgsfaktoren des Pjm • Besondere Aspekte beim Management von Softwareprojekten • Vorgehensmodelle zur Abwicklung von Softwareprojekten: klassische Modelle, Agile Methoden: Prinzipien, Agile Frameworks und Agile Praktiken • Projektorganisation, Aufgaben und Rollen im IT-Projektmanagement • Werkzeuge für das IT-Projektmanagement • Anforderungsmanagement, Lasten- und Pflichtenheft, Methoden für Aufwandsschätzungen, Projektplanung/Arbeitspakete, Termin- und Zeitplanung • Ressourcenmanagement, Kostenplanung, Projektcontrolling, Projektsteuerung • Qualitätssicherung in Softwareprojekten, Risikoanalyse/ Risikomanagement • Projektkommunikation intern/extern, Projektdokumentation • Teamführung, Dynamik in Teams, Umgang mit Konflikten • Einsatz von KI-Tools in Projekten 				
4	Lehr- und Lernformen Vorlesung, begleitend Analyse von Fallbeispielen, praktische Übungen zum Einsatz von Tools, Rollen-/ Planspiele				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Kombinationsprüfung nach § 22 RPO (Klausur und Hausarbeit mit Fachvortrag) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Kuster et al., Handbuch Projektmanagement: agil - klassisch - hybrid, Springer Berlin, 2022. Tiemeyer, Handbuch IT-Projektmanagement: Vorgehensmodelle, Managementinstrumente, Good Practices, Hanser 2023. weitere Literatur und Quellen zu Einzelthemen

Pflichtmodule des 5. Semesters

Webentwicklung Backend

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		180h	6 CP	5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können das Backend einer einfachen Webanwendung erstellen und dabei sowohl die funktionalen Aspekte (Umsetzung von Use-Cases, Datenpersistenz) als auch nicht-funktionale Aspekte (Skalierbarkeit, Schutz vor gängigen Angriffen wie XSS und XSRF) angemessen berücksichtigen. Sie <ul style="list-style-type: none">verstehen den Einfluss der Anwendungsarchitektur auf Skalierbarkeit und Sicherheit einer Webanwendung und die Bedeutung eines Zonenmodells,können mithilfe von Werkzeugen wie OpenAPI zu einem gegebenen Anwendungsfall ein API entwerfen,können ein solches API implementieren,können geeignete Authentisierungsverfahren für eine Webanwendung auswählen und implementieren, undkönnen geeignete Strategien für den Test und das Deployment einer Webanwendung entwickeln.					
3	Inhalte <ul style="list-style-type: none">Serverseitige Frameworks wie z.B. NodeJS und ExpressAnbindung von Datenbanken in WebanwendungenSpezifikation von REST-APIsAuthentisierungsverfahren in WebanwendungenTest von REST-APIsDeployment-Optionen für Webanwendungen					
4	Lehrformen Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen Formal: „Webentwicklung Frontend“ Inhaltlich:					
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistungen keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					

9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl: Springer, Sebastian: Node.js: das Praxisbuch, Bonn 2016 Bojinov, Valentin: RESTful web API design with Node.js, Birmingham 2016

Softwareengineering

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 5. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Modul Softwareengineering führt auf die systematische Erstellung von Softwareprodukten hin. Es werden die dazu gängigen Techniken in Modellierung, Projektmanagement, Entwicklung vorgestellt und angewandt.				
3	Inhalte <ul style="list-style-type: none">• Überblick über Historie der Softwareentwicklung Spiralmodell, VModell und Prototyping Ansätze• Vorgehensmodell mit<ul style="list-style-type: none">• Requirements Engineering• Anforderungsanalyse• iterativ inkrementelle Komponentenentwicklung• Systemtest• Lasten- und Pflichtenheft• Prototyping• Modellierung statischer und dynamischer Softwareaspekte• Erstellung objektorientierter Software• Anwendung der Unified Modelling Software• Patterns• Softwarearchitekturen• Dokumentation• Systematisches Testen				
4	Lehr- und Lernformen Vorlesungen, seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur oder Mündliche Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Uwe Klug
12	Sonstige Informationen Literaturauswahl: Balzert, Helmut: <i>Lehrbuch der Software-Technik, Entwurf, Implementierung, Installation und Betrieb</i> , Spektrum Akademischer Verlag, Balzert, Helmut: <i>Lehrbuch der Software-Technik, Basiskonzepte und Requirements Engineering</i> , Spektrum-Verlag Helmut Balzert; Heide Balzert; <i>Lehrbuch der Objektmodellierung</i> ; Spektrum Verlag Joachim Goll: <i>Methoden und Architekturen der Softwaretechnik</i> , Vieweg + Teubner Joachim Goll: <i>Architektur- und Entwurfsmuster der Softwaretechnik</i> , Springer Vieweg Ian Sommerville; <i>Softwareengineering</i> ; Pearson Studium Requirements Engineering; Chris Rupp; Hanser Fachbuch Spillner, A. / Lenz, T.; <i>Basiswissen Softwaretest</i> ; dpunkt Verlag

Pflichtmodule des 6. Semesters

Projektarbeit

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		240 h	9 CP	6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 1 SWS / 11,75 h seminaristischer Unterricht b) 3 SWS / 33,25 h Projektarbeit		Kontaktzeit 4 SWS / 45 h	Selbststudium 225 h		geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können Methoden der systematischen Softwareentwicklung innerhalb einer größeren Aufgabenstellung anwenden.					
3	Inhalte Es wird ein Projekt bearbeitet, in dem die erlernten Techniken des Softwareengineerings einschließlich Vorgehensmodell und Projektmanagement angewendet werden. Die Aufgaben sollen aus einem praxisnahem Umfeld stammen, z.B. aus dem hochschulinternen Laborbetrieb oder aus einem Industriebetrieb. Bei Interesse kann hier die Einarbeitung in ein Themengebiet für eine Bachelorarbeit erfolgen.					
4	Lehr- und Lernformen Seminaristischer Unterricht, Projektarbeit					
5	Teilnahmevoraussetzungen Formal: keine Inhaltlich: Modulen Java-Programmierung 1 und 2, Softwareengineering 1 und Datenbanken 1 und 2					
6	Prüfungsformen Schriftliche Ausarbeitung mit Präsentation (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 9/180 = 5%					

11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Uwe Klug
12	Sonstige Informationen Literaturauswahl: Helmut Balzert; Lehrbuch der Software Technik I+II; Spektrum Verlag Helmut Balzert; Lehrbuch Grundlagen der Informatik; Spektrum Verlag Heide Balzert; Lehrbuch der Objektmodellierung; Spektrum Verlag W. Zuser u.a.; Softwareengineering; Pearson Studium Ian Sommerville; Softwareengineering; Pearson Studium Requirements Engineering; Chris Rupp; Hanser Fachbuch

Bachelorarbeit

Kennnummer		Workload 360 h	Credits 12 CP	Studiensemester 6. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 8 Wochen
1	Lehrveranstaltungen		Kontaktzeit	Selbststudium	geplante Gruppengröße alle	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sind befähigt, innerhalb einer vorgegebenen Frist eine praxisorientierte Aufgabe aus dem Bereich der Anwendungsentwicklung oder Systemintegration selbstständig mit den in der Anwendung erprobten wissenschaftlichen und fachpraktischen Methoden zu bearbeiten und in fachübergreifende Zusammenhänge zu stellen.					
3	Inhalte Es soll ein in der Regel ein praxisorientiertes Problem aus dem Bereich der Anwendungsentwicklung oder Systemintegration mit den im Studium erlernten wissenschaftlichen und fachpraktischen Methoden in begrenzter Zeit unter Anleitung eines erfahrenen Betreuers gelöst werden. Die Bachelorarbeit ist entweder eine eigenständige Ausarbeitung oder betrachtet ein bekanntes Thema unter neuen Aspekten.					
4	Lehr- und Lernformen Angeleitetes, eigenverantwortliches Arbeiten					
5	Teilnahmevoraussetzungen siehe § 20 FPO					
6	Prüfungsformen schriftliche Ausarbeitung mit Präsentation					
7	Prüfungsvorleistung					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 12/180 = 6,67%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Alle Dozent*innen des Bachelorstudiengangs Informatik					
12	Sonstige Informationen					

Kolloquium

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	90 h	3 CP	6. Sem.	Sommersemester	30-45 Min.
1	Lehrveranstaltungen	Kontaktzeit	Selbststudium	geplante Gruppengröße alle	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sind befähigt, die Ergebnisse der Bachelorarbeit, ihre fachlichen Grundlagen, ihre fachübergreifenden Zusammenhänge und ihre außerfachlichen Bezüge mündlich darzustellen und selbstständig zu begründen sowie ihre Bedeutung für die Praxis einzuschätzen.				
3	Inhalte Zunächst wird der Inhalt der Bachelorarbeit aus dem Bereich der Anwendungsentwicklung oder Systemintegration im Rahmen eines Vortrages präsentiert. Anschließend sollen in einer Diskussion Fragen zum Vortrag und zur Bachelorarbeit beantwortet werden.				
4	Lehr- und Lernformen				
5	Teilnahmevoraussetzungen siehe § 22 FPO				
6	Prüfungsformen Präsentation und mündliche Prüfung				
7	Prüfungsvorleistung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				
10	Stellenwert der Note für die Endnote 3/180 = 1,67%				
11	Modulbeauftragte*r und hauptamtlich Lehrende Alle Dozent*innen des Bachelorstudiengangs Informatik				
12	Sonstige Informationen				

Wahlpflichtmodule

Betriebswirtschaftslehre

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 3./5. Sem.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden sollen Grundbegriffe (Umsatz, Gewinn, Rentabilitäten, Produktivität etc.) definieren und auf unternehmerische Sachverhalte anwenden können. Ferner sollen die Studierenden Kenntnisse zum organisatorischen Aufbau von Unternehmen (Einlinien-, Stablinien- sowie Spartensystem) und zu den Rechtsformen (OHG, KG, AG, GmbH) erwerben. Darüber hinaus sollen die Studierenden Instrumente und Maßnahmen aus den Funktionsbereichen der Unternehmen kennen lernen, wie z.B. die ABC-Analyse, die Bestellmengenrechnung, Marketingmaßnahmen zur Verbesserung der Verkaufssituation (Werbung, Preisfindung usw.). Die Studierenden erhalten die Kompetenz, wirtschaftliche Gegebenheiten in Unternehmen besser verstehen und beurteilen zu können. Detaillierte Lernziele werden im Rahmen der Veranstaltung bekannt gegeben.</p>				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen: Grundbegriffe, Unternehmensziele • Unternehmen: Organisation, Rechtsformen, Sozialpartner • Beschaffung: Beschaffungsplanung, Investitionsrechnung • Marketing: Markt, Preisbildung 				
4	Lehr- und Lernformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) im Fachbereich Maschinenbau				
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$				

11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Lothar Winnen
12	Sonstige Informationen Literaturauswahl: Jung, H.: Allgemeine Betriebswirtschaftslehre, München/Wien Schierenbeck, H.: Grundzüge der Betriebswirtschaftslehre, München/Wien Thommen, J.-P./Achleitner, A.-K.: Allgemeine Betriebswirtschaftslehre. Umfassende Einführung aus managementorientierter Sicht, Wiesbaden Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre, München

Cloud Computing

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Wahlprüfungsmodul der Studiengattung Systemintegration						
Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierende verstehen die Prinzipien verteilter Systeme und können Cloud-Computing Lösungen hinsichtlich ihrer technischen und ökonomischen Vorteile evaluieren. Sie sind befähigt, die Eignung verschiedener Cloud-Modelle für spezifische Anwendungsfälle zu diskutieren und kennen unterschiedliche Serviceangebote. Durch den Umgang mit Web- und API-Schnittstellen können sie Cloud-Dienste effizient nutzen. Die Studierenden sind in der Lage, die Konzepte für den Aufbau und das Management von Public-Cloud-Umgebungen zu entwerfen. Die im Rahmen der Praktika erworbenen Erfahrungen mit einem führenden Cloud Provider befähigen die Studierenden dazu, ihr Wissen adaptiv auf diverse andere Cloud-Service-Plattformen anzuwenden.					
3	Inhalte <ul style="list-style-type: none">• Technologische Grundlagen von Cloud Computing (Virtuelle Maschinen, Netzwerk-Virtualisierung, Container, etc.)• Cloud Service Ebenen• Beispiele für Cloud Dienste• Cloud APIs• Cloud Storage• Entwurf Cloud-basierter Anwendungen, Microservices• Serverless Computing					
4	Lehrformen Vorlesung, Übung					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Betriebssysteme, Virtualisierung					
6	Prüfungsformen Mündliche Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistungen keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: Ian Foster und Dennis B. Gannon: Cloud Computing for Science and Engineering, MIT Press, 2017 Thomas Erl, Robert Cope und Amin Naserpour: Cloud Computing Design Patterns, Prentice Hall, 2015 Oliver Liebel: Skalierbare Container-Infrastrukturen: Das Handbuch für Administratoren und DevOps-Teams, Rheinwerk, 2023 Dan C. Marinescu: Cloud Computing: Theory and Practice, Morgan Kaufmann, 2017 Edouard Bugnion, Jason Nieh, und Dan Tsafir: Hardware and Software Support for Virtualization, Morgan & Claypool Publishers, 2017 John J. Geewax: Google Cloud Platform in Action, Manning, 2018 Valliappa Lakshmanan: Data Science on the Google Cloud Platform, O'Reilly, 2022 Dan Sullivan, Google Cloud Certified Associate Cloud Engineer Study Guide, Wiley & Sons, 2023

Datenbanken 2

(Wahlpflichtmodul der Studienrichtungen Anwendungsentwicklung und Künstliche Intelligenz)

Wahlmodul der Studiengängen Anwendungsentwicklung und Künstliche Intelligenz					
Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180 h	6 CP	3./5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Modul Datenbanken 2 soll Kenntnisse in der Datenmodellierung ergänzen, detaillierte Kenntnisse über anspruchsvolle, z.B. mengenbasierte, SQL-Anweisungen vermitteln. Ferner soll in die Handhabung Schnittstellen von relationalen Datenbanken zu Programmierungsumgebungen vertieft werden.				
3	Inhalte <ul style="list-style-type: none">• Modellierung von Datenbanken• mengenalgebraische Grundlagen und Verbundanweisungen• Unterabfragen<ul style="list-style-type: none">○ skalare Unterabfragen, Listenabfragen○ Unterabfragen in Ergebnisspalten○ Unterabfragen als Datenquellen○ Zusammenhang Unterabfragen und Gruppierungen○ Unterabfragen zur Wertebereichseinschränkung• mengenmäßige Weiterverarbeitung von Selektionsergebnissen• Umgang mit Benutzerrechten in Form von Einzelberechtigungen und Benutzerrollen• Benutzersichten auf Datenbestände• Programmierung von DB-Anwendungen<ul style="list-style-type: none">○ Umgang mit Stored Procedures○ Verwendung von Triggern○ Einsatz von Transaktionen• Ausblick auf alternative Datenbankkonzepte				
4	Lehr- und Lernformen Vorlesungen, seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: „Grundlagen der Informatik1 und 2“, „Einführung in die Programmierung“ Inhaltlich: grundlegende Kenntnisse relationaler Datenbanken und SQL				
6	Prüfungsformen Klausur und Kombination mit Vorleistung aus dem Praktikum/seminaristischem Unterricht oder prozessorientierte Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Uwe Klug
12	Sonstige Informationen Literaturauswahl: R. Elmasri, S. B. Navathe, Grundlagen von Datenbanksystemen, Addison-Wesley Heuer, G. Saake, Datenbanken: Konzepte und Sprachen, Internat. Thomson Publ. U. Klug; SQL - Der Einstieg in die deklarative Programmierung; W3L Verlag U. Klug; Datenbankanwendungen entwerfen & programmieren; W3L Verlag G.Lausen; Datenbanken; Spektrum Akademischer Verlag G.Saake, K.-U. Sattler; Datenbanken & Java; dpunkt.Verlag St. Edlic; NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken; Hanser Verlag M. Kaufmann, A. Meier; SQL- & NoSQL-Datenbanken; Springer Verlag

Datenschutz

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 3.- 6. Semester	Häufigkeit Winter- /Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen 2 SWS / 22,5 h Vorlesung 2 SWS / 22,5 h Seminar	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße alle	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden lernen, in der gesetzlich vorgeschriebenen Weise mit personenbezogenen Daten umzugehen. Gleichzeitig stellt dies eine Einführung in die IT-Sicherheit dar.				
3	Inhalte Der Kurs vermittelt Kenntnisse in allen wesentlichen Bereichen des privaten Datenschutzes und zerfällt in einen rechtlichen und einen technischen Teil. rechtlicher Teil <ul style="list-style-type: none">einschlägige Gesetze: Bundesdatenschutzgesetz, aber auch Sondergesetze wie zum Beispiel das Telemediengesetz, Telekommunikationsgesetzes, Sozialgesetzbuch und verwandte GesetzeGrundzüge des Datenschutzes: Definitionen, Datensparsamkeit, DatenvermeidungÜbermittlung von Daten zwischen verschiedenen Stellen, Übermittlung von Daten ins Ausland, Umgang mit Daten im InternetErlaubnistatbestände der Datenerhebung, -verarbeitung und -nutzungGrenzen im Bereich des Datenschutzes im ArbeitsverhältnisBetrieblichen Datenschutzbeauftragte: Bestellung, Aufgaben, Abberufung und Kündigung				
4	Lehr- und Lernformen Vorlesung, Seminar				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				
10	Stellenwert der Note für die Endnote 6/180 = 3,33%				
11	Modulbeauftragte*r und hauptamtlich Lehrende RA Prof. Andreas Göbel				
12	Sonstige Informationen				

Deep Learning

(Wahlpflichtmodul der Studienrichtung Künstliche Intelligenz)

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180 h	6 CP	5. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die theoretischen Grundlagen künstlicher neuronaler Netze und verstehen, wie man künstliche neuronale Netze in der Praxis verwendet. Sie sind in der Lage, Aufgaben wie die Erkennung von Ziffern oder die Klassifikation von Bildern mithilfe von Deep-Learning Bibliotheken selbstständig zu lösen.				
3	Inhalte <ul style="list-style-type: none"> • Deep Learning und Künstliche neuronale Netze (KNN) • Trainingsalgorithmen für KNN • Bewertungsverfahren für die Modellgüte • Spezielle Typen von KNN: Convolutional Neural Networks, Recurrent Neural Networks • Training von CNNs • Training von RNNs mit LSTM • Werkzeuge für Deep Learning • Praktische Beispiele: MNIST Datensatz, ImageNet 				
4	Lehr- und Lernformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Formal: „Einführung Machine Learning“ Inhaltlich: Modul Skriptsprachen (Python-Kenntnisse)				
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Prof. Dr. Christian Gawron				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: Patterson, Josh; Deep learning: a practitioner's approach, O'Reilly 2017 Chollet, Francois; Deep Learning with Python, 2018 Wartala, Ramon; Praxiseinstieg Deep Learning: mit Python, Caffé, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen, O'Reilly 2018

Effiziente Algorithmen

(Wahlpflichtmodul der Studienrichtungen Anwendungsentwicklung und Künstliche Intelligenz)

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180h	6 CP	4./6. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 18 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen verschiedene fortgeschrittene Methoden für den Entwurf von Algorithmen und können diese zielgerichtet für die Lösung von Aufgaben aus unterschiedlichen Problembereichen auswählen und einsetzen. Sie kennen die Unterschiede zwischen verschiedenen Entwurfsstrategien und ihre Auswirkungen auf den Zeit- und Platzbedarf des Algorithmus. Die Studierenden können bei Dynamischer Programmierung Speicheroptimierungen vornehmen, um diese auch platzeffizient zu implementieren. Sie kennen das Konzept der NP-Vollständigkeit und können einfache Reduktionen nachvollziehen.				
3	Inhalte <ul style="list-style-type: none"> Entwurf von Algorithmen (Vertiefung / fortgeschrittene Methoden) <ul style="list-style-type: none"> Dynamische Programmierung Greedy Algorithmen Approximationsalgorithmen fortgeschrittene Analyse von Algorithmen, grundlegende Komplexitätsklassen <ul style="list-style-type: none"> Ressourcen / Effizienz, amortisierte Analyse Einführung in P vs NP, Vollständigkeit/Reduktion Anwendungen (exemplarisch) <ul style="list-style-type: none"> z.B. fortgeschrittene Graphalgorithmen, Optimierungsprobleme (Knapsack, TSP, Bin Packing), Algorithmische Geometrie, Matrix-Operationen, Randomisierte Algorithmen, ...) 				
4	Lehr- und Lernformen Vorlesung / seminaristischer Unterricht (50%), Praktikum (50%)				
5	Teilnahmevoraussetzungen Formal: „Mathematik 1 und 2“ Inhaltlich: „Mathematik 1 und 2“, Programmierkenntnisse aus den ersten beiden Semestern				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung: Bearbeitung von Programmieraufgaben und Aufgaben in Moodle - die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr.-RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Vöcking, Taschenbuch der Algorithmen. Springer Cormen, Leiserson, Rivest, Stein, Algorithmen - Eine Einführung, Oldenbourg Wissenschaftsverlag Ottmann, Widmayer, Algorithmen und Datenstrukturen, Springer Al-Taie, Python for Graph and Network Analysis, Springer Nature

Einführung in die Theoretische Informatik

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180 h	6 CP	4./6. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 4 SWS / 45 h Seminar		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße 10-15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen und verstehen wesentliche Konzepte und Methoden zur theoretischen Modellierung von Berechenbarkeit durch verschiedene Automaten-/Maschinenmodelle. Sie kennen die Stärken und Grenzen dieser Modelle, sowie deren Beziehungen untereinander. Die Studierenden wenden die Formalisierungen und abstrakten Konzepte sicher an, sie sind geübt im Anwenden der mathematischen Arbeitsweisen in der theoretischen Informatik. Für konkrete praktische Aufgaben können sie geeignete Modelle auswählen und Werkzeuge/Methoden der theoretischen Informatik darauf anwenden. Die Studierenden sind außerdem dazu befähigt ihr eigenes Wissen zu Theoretischer Informatik selbstständig mit Hilfe von Lehrbuchtexten zu erweitern.					
3	Inhalte <ul style="list-style-type: none">• mathematische Grundlagen und Logik, inkl. Einführung in Beweismethoden der Theoretischen Informatik• Formale Sprachen und Chomsky-Hierarchie• Automaten-/Maschinenmodelle unterschiedlicher Komplexität (endliche Automaten, Kellerautomaten, Turing-Maschinen)• Ausgewählte Kapitel der Berechenbarkeitstheorie und Entscheidbarkeits-/ Unentscheidbarkeitsresultate• Einführung in Komplexitätsklassen, P/NP, NP-Vollständigkeit, Approximationsalgorithmen					
4	Lehr- und Lernformen Flipped Classroom mit Lernteamcoaching, ergänzt durch Bearbeitung theoretischer und praktischer Aufgaben auf Papier und mit Lernsoftware (Übung/ Praktikum), angeleiteter Projektarbeit in Kleingruppen, Projektpräsentationen					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: „Mathematik (1 und 2)“, „Einführung in die Programmierung“, „Objektorientierte Programmierung“					
6	Prüfungsformen Kombinationsprüfung nach §19 BPO: schriftliche Ausarbeitung (§18) kombiniert mit zusätzlicher Klausur (§15) (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Dirk W. Hoffmann, Theoretische Informatik, Hanser Gottfried Vossen, Kurt-Ulrich Witt, Grundkurs Theoretische Informatik, Springer Verlag Michael R. Garey, David S. Johnson, Computers and intractability: a guide to the theory of NP-completeness, Freeman Juraj Hromkovič, Theoretische Informatik, Springer Verlag Ingo Wegener, Theoretische Informatik- eine algorithmenorientierte Einführung, Teubner Katrin Erk, Lutz Priese, Theoretische Informatik, Springer Verlag

Einführung Machine Learning

(Wahlpflichtmodul der Studienrichtung Anwendungsentwicklung)

Prüfungsleistung der Studiengänge / Anwendungsorientierung					
Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180 h	6 ECTS	4. / 6. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen grundlegenden Machine Learning Methoden zur Regression und Klassifikation und können diese im Gebiet der künstlichen Intelligenz einordnen. Sie kennen Gütekriterien für binäre Klassifikatoren und diese angemessen anwenden. Für einfache Problemstellungen können Sie geeignete Machine Learning Verfahren auswählen, Lösungsansätze entwickeln und diese mithilfe der Programmiersprache Python sowie entsprechender Bibliotheken umsetzen und evaluieren.				
3	Inhalte <ul style="list-style-type: none">• Übersicht über das Themenfeld KI und Einordnung der Methoden des Maschine Learnings• Python Bibliotheken für numerisches Rechnen, Verarbeitung strukturierter Daten, Maschinelles Lernen und Visualisierung• Grundlagen der Statistik• Überblick von Klassifikationsverfahren und Gütekriterien• Lineare Regression• Trainieren von Machine Learning Modellen• Binäre und multinomiale logistische Regression• Entscheidungsbäume• Support Vector Machines• Neuronale Netzwerke				
4	Lehr- und Lernformen Vorlesung (50%) und Praktikum (50%)				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Skriptsprachen oder grundlegende Programmierkenntnisse in Python				
6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: Ethem Alpaydin: <i>Maschinelles Lernen</i> . De Gruyter Oldenburg, 2. Edition, 2019 Aurélien Géron: <i>Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow</i> . O'Reilly, 2017 Sebastian Raschka, Vahid Mirjalili: <i>Machine Learning mit Python und Scikit-Learn und TensorFlow</i> . mitp, 2017 Jörg Frochte: <i>Maschinelles Lernen: Grundlagen und Algorithmen in Python</i> , Hansa 2019

Fortgeschrittene Internettechnologien

(Wahlpflichtmodul der Studienrichtung Anwendungsentwicklung)

Wahlmodul der Studiengänge Anwendungsentwicklung						
Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		180 h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen		Kontaktzeit	Selbststudium	geplante Gruppengröße	
	a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		4 SWS / 45 h	135 h	b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen					
	Die Vorlesung vermittelt fortgeschrittene Kenntnisse zur Erstellung professioneller Webanwendungen mit der Jakarta Enterprise Edition. Darüber hinaus wird der Einsatz ausgewählter Frameworks und die Anbindung von Datenbanken und Applikationsservern erlernt.					
3	Inhalte					
	Der Inhalt der Vorlesung umfasst die Themen					
	<ul style="list-style-type: none">• Servlets• Java Server Pages• Frameworks zur Erstellung von Webapplikationen• Anbindung von Datenbanken und Applikationsservern					
	Im Praktikum werden auf die Vorlesung abgestimmte Präsenzaufgaben bearbeitet. Dabei wird sukzessive ein einfacher Prototyp einer Webanwendung (Online-Auktion, Bulletin Board etc.) erstellt.					
4	Lehr- und Lernformen					
	Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen					
	Formal:					
	Inhaltlich: Module Grundlagen der Informatik, Objektorientierte Programmierung und Datenbanken					
6	Prüfungsformen					
	Klausur					
	(Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung					
	Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten					
	Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Andreas Steins
12	Sonstige Informationen Literaturauswahl: Hans Bergsten, JavaServer Faces. Building Web-based User Interfaces, O'Reilly Michael C. Daconta et al, More Java Pitfalls: 50 New Time-Saving Solutions and Workarounds, Wiley Marty Hall, Core Servlets and Java Server Pages, SUN Microsystems Press, Prentice Hall, Online unter http://pdf.coreservlets.com/ Marty Hall, More Servlets and Java Server Pages, SUN Microsystems Press, Prentice Hall, Online unter http://pdf.moreservlets.com/ Jason Hunter, Java Servlet Programming, O'Reilly Kito Mann, JavaServer Faces in Action, Manning

Funktionale Programmierung

(Wahlpflichtmodul der Studienrichtung Anwendungsentwicklung)

Wahlpflichtmodul der Studiennichtung Anwendungsentwicklung						
Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h		geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Vorlesung vermittelt grundlegende Kenntnisse der funktionalen Programmierung mit Hilfe eines modernen Lisp-Dialekts.					
3	Inhalte Der Inhalt der Vorlesung umfasst die Themen <ul style="list-style-type: none">• Das Programmierparadigma der funktionalen Programmierung• Die Programmiersprache Clojure Im Praktikum werden auf die Vorlesung abgestimmte Präsenzaufgaben bearbeitet.					
4	Lehr- und Lernformen Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Module Grundlagen der Informatik und Objektorientierte Programmierung.					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Andreas Steins					
12	Sonstige Informationen Literaturauswahl: Im Web: https://clojure.org/guides/getting_started					

Frontend-Frameworks für Webanwendungen

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden verstehen die Funktionsweise moderner Single-Page-Applications und kennen deren Vor- und Nachteile in Bezug auf Usability und Sicherheit. Sie kennen aktuelle Frameworks zur Entwicklung solcher Anwendung und können mit deren Hilfe eigene Anwendungen entwickeln. Im Praktikum erwerben die Studierenden die notwendigen Kompetenzen für das Design, die Entwicklung, den Test und das Deployment von Single-Page-Applications.				
3	Inhalte <ul style="list-style-type: none">JavaScript-basierte Frontend-Frameworks<ul style="list-style-type: none">Vue.jsReactAngularTools für den Test von Webanwendungen<ul style="list-style-type: none">Cypress.ioBuild- und Deployment von Webanwendungen<ul style="list-style-type: none">WebpackCI mit GitHub ActionsDeployment mit Docker und traefik				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Hausarbeit (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistungen keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33 %
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl: Oliver Zeigermann / Nils Hartmann <i>React: Grundlagen, fortgeschrittene Techniken und Praxistipps – mit TypeScript und Redux</i> Steyer, Ralph <i>Webanwendungen erstellen mit Vue.js: MVVM-Muster für konventionelle und Single-Page-Webseiten</i> Ferdinand Malcher / Johannes Hoppe / Danny Koppenhagen Angular : <i>Grundlagen, fortgeschrittene Themen und Best Practices – inklusive NativeScript und NgRx</i>

Gender und Diversity in der Informatik

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 3. / 5.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen 4 SWS / 22,5 h Vorlesung / Seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Gender und Diversity finden in der Informatik in Industrie und Hochschule verstärkt Beachtung, denn ohne Vielfalt kann Informatik ihrer Verantwortung im Digitalisierungsprozess nicht gerecht werden. Die Teilnehmenden kennen die grundlegenden Konzepte und empirischen Ergebnisse zu Gender und Diversity in Bezug auf Informatik und Technikgestaltung. Sie können Algorithmen, IT-Systeme und IT-Prozesse in Hinsicht auf Gender und Diversity selbständig analysieren. Sie kennen grundlegende Methoden der Gender- und Diversity-reflektierten Gestaltung von IT und können diese praktisch auf die eigene Arbeit in der Informatik anwenden.				
3	Inhalte <ul style="list-style-type: none">Grundlagen zu Gender und Diversity in Informatik und Digitalen Technologien<ul style="list-style-type: none">Dimensionen von Diversity, Intersektionalitätaktuelle Situation zu Gender und Diversity in der InformatikFachkultur in Studium und Arbeitsleben, Rollenmodelle, MentoringBeachtung von Gender und Diversity in der Gestaltung von IT<ul style="list-style-type: none">Praxisbeispiele, empirische BefundeMethoden für G&D in der Informatik: IT-Design for all / GERD-ModellErkennung und Vermeidung von Bias in Algorithmen / IT-ProzessenAnwendung Gender- und Diversity-reflektierter Methoden auf ein konkretes Projekt				
4	Lehr- und Lernformen Vorlesung / Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Hausarbeit mit Fachvortrag (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literatúrauswahl (wird jeweils um aktuelle Themen ergänzt): Johnson, Toward Information Justice. Springer, 2018. Rudolph et al. (Hrsg.), Geschlechtergerechtigkeit und MINT, Budrich, 2022. Benjamin, Race after Technology: Abolitionist Tools for the New Jim Code. Polity, 2019. Kerkmann/ Lewandowski. Barrierefreie Informationssysteme: Zugänglichkeit für Menschen mit Behinderung in Theorie und Praxis. De Gruyter, 2015. Apelt et al. (Hrsg.), Buzzword Digitalisierung - Relevanz von Geschlecht und Vielfalt in digitalen Gesellschaften, Budrich, 2021. Browne et al. (Hrsg.), Feminist AI. Critical Perspectives on Algorithms, Data, and Intelligent Machines, Oxford 2023. D'Ignazio/Klein, Data Feminism, MIT Press. 2020. Umoja Noble et al. (Hrsg.), The Intersectional Internet, Peter Lang 2018 Leicht-Scholten et al. (Hrsg.), Informatikkultur neu denken, Integration von Gender und Diversity in MINT-Studiengänge, Springer 2014 ACM Conference on Gender & IT, Proceedings, ACM Press.

Hardwarenahe Programmierung

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180	6 CP	4./6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Nach dem erfolgreichen Abschluss des Moduls sind die Studierenden in der Lage, effiziente Programme in der Programmiersprache C zu entwickeln. Sie beherrschen den Umgang mit Zeigern und der dynamischen Speicherverwaltung in C, was Ihnen erlaubt, komplexe Datenstrukturen und speicherintensive Anwendungen effizient zu implementieren. Die Studierenden sind mit den Methoden zum Debuggen und Testen von C-Programmen vertraut und können diese anwenden, um robuste und fehlerfreie Software zu erstellen. Sie haben die Verwendung von integrierten Entwicklungsumgebungen (IDEs) speziell für eingebettete Systeme anhand von praktischen Beispielen erprobt. Die Studierenden sind in der Lage, Hardwarekomponenten direkt anzusteuern und zu regeln, indem Sie Programme schreiben, die mit verschiedenen Sensoren und Aktoren interagieren.				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen der Programmiersprache C: Aufbau und Struktur eines C-Programms, Operatoren und Datentypen, Kontrollstrukturen (Schleifen, Verzweigungen), Funktionen und Modularisierung • Zeiger und dynamische Speicherverwaltung: Speicherallokation und -freigabe • Verwendung von Datenstrukturen: Arrays und Strings, structs, Listen, etc. • Umgang mit Debugging-Tools und Testmethoden • Nutzung von IDEs für eingebettete Systeme • Betriebssysteme für eingebettete Systeme • Ein- und Ausgabe mit Sensoren und Aktoren • Praktische Anwendungen 				
4	Lehrformen Vorlesung (50%), Praktikum (50%)				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Mündliche Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: Brian W. Kernighan, Dennis M. Ritchie: <i>The C Programming Language</i> , 2nd Edition, Prentice Hall, 1988. Michael J. Pont: <i>Embedded C</i> (2nd Edition), Addison-Wesley, 2002. James W. Grenning: <i>Test-Driven Development for Embedded C</i> , O'Reilly, 2011. Patrick Ritschel: <i>Embedded Systems mit RISC-V und ESP32-C3: eine praktische Einführung in Architektur, Peripherie und eingebettete Programmierung</i> , dpunkt.verlag, 2023. <i>Zephyr Project Documentation</i> , docs.zephyrproject.org

IT-Recht

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die Rechte und Pflichten bei dem Erwerb von Hardware und Software sowie bei der Betätigung im Internet. Sie wissen, wo die besonderen Gefahren liegen und wie man sie vermeidet.					
3	Inhalte <ul style="list-style-type: none">• Einführung in das Recht von EDV und Internet• typische Probleme beim Kauf von Hardware und Software• typische Probleme bei der Erstellung von SW und der Durchführung von Dienstleistungen• Allgemeine Geschäftsbedingungen: Vereinbarung und zulässige Inhalte• Das EDV-Projekt: typische Probleme und Fallen• Grundzüge des Urheberrechts: Lizenzen• EDV-Recht im Arbeitsverhältnis: Abmahnung, Kündigung, Beweislast• Vertragsschluss im Internet• Typische Verträge im Internet: Versteigerung, Power-shopping u.a.• e-commerce: online-Handel und Verbraucherschutz• Haftung für Inhalte und Links im Internet: Access- und Contentprovider• Internet und Email am Arbeitsplatz• Grundzüge des Rechts der Domains• Datenschutz I: Grundzüge• Datenschutz II: Online-Dienste, Übermittlung ins Ausland• Strafrecht und Compliance: Computerstraftaten und Terrorbekämpfung					
4	Lehr- und Lernformen Vorlesung, Übung					
5	Teilnahmevoraussetzungen Formal: Inhaltlich:					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Rechtsanwalt Andreas Göbel					
12	Sonstige Informationen					

Marketing

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180 h	6 CP	4. / 6. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sollen handlungsorientiert in das Fach Marketing bzw. Industriegütermarketing eingeführt werden. Sie sollen mit den Fachtermini des Industriegütermarketings vertraut gemacht werden und lernen, wie die Absatzsituation eines Unternehmens ermittelt wird, welche Möglichkeiten (Absatzpolitiken) ein Unternehmen hat, seine Absatzsituation hinsichtlich eines vorgegebenen Unternehmensziels zu verbessern. Detaillierte Lernziele werden im Rahmen der Veranstaltung bekannt gegeben.				
3	Inhalte <ul style="list-style-type: none"> • Marketingbegriff • Besonderheiten im Industriegütermarketing • Nachfrageanalyse • Konkurrenzanalyse • Marketingstrategien • Kaufentscheidungstypen • Marketing im Produkt-/Zuliefergeschäft 				
4	Lehr- und Lernformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Im Studiengang Angewandte Biologie sowie in Studiengängen des Nachbarfachbereichs Madschinenbau				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Lothar Winnen
12	Sonstige Informationen Literaturauswahl: Backhaus, K./Voeth, M.: Industriegütermarketing, München Bruhn, M.: Marketing. Grundlagen für Studium und Praxis, Wiesbaden Meffert, H./Burmann, C./Kirchgeorg, M.: Marketing. Grundlagen marktorientierter Unternehmensführung. Konzepte - Instrumente – Praxisbeispiele, Wiesbaden

Mobile Applikationen

(Wahlpflichtmodul der Studienrichtung Anwendungsentwicklung)

Wahlprüfung aus der Studiengang Anwendungsentwicklung					
Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180 h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sind in der Lage, mobile Applikationen auf Basis der gängigen vier Entwicklungsparadigmen zu entwickeln: Mobile Web-Apps, hybride Apps, Cross-Plattform-Apps sowie native Apps. Sie haben insbesondere vertiefte Kenntnisse über die Architektur der Android-Plattform erworben, und sie beherrschen den Umgang mit der Entwicklungsumgebung Android Studio. Den Einsatz von Android-App-spezifischen Elementen wie etwa Intents haben sie eingeübt, und sie können bei Konzeption und Implementierung die Besonderheiten von mobilen Anwendungen berücksichtigen. Im Zuge der Arbeit mit Activity Layout Designs haben die Studierenden auch den Umgang mit der Auszeichnungssprache XML vertieft.				
3	Inhalte <ul style="list-style-type: none">• Paradigmen für die Entwicklung mobiler Endgeräte• Mobile Web-Anwendungen auf Basis des Ionic- und Vue.js-Frameworks• Native Entwicklung mobiler Anwendungen mit Android<ul style="list-style-type: none">○ Architektur der Android-Plattform○ Activities und ihr Lebenszyklus○ Views und Layouts○ Intents○ Menüs und Navigation○ Maps (Google und Open Street Map)○ Sensoren○ Bar- und QR-Codes				
4	Lehr- und Lernformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Softwaretechnische Grundkenntnisse (Java Standard Edition, C++)				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) im Verbund.B.Sc.-Studiengang Angewandte Informatik
10	Stellenwert der Note für die Endnote $6 / 180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Doga Arinir
12	Sonstige Informationen Literaturauswahl: Arinir, D.; Mobile Computing, Springer Vieweg, 978-3-662-67412-3, 20232

Natural Language Processing

(Wahlpflichtmodul der Studienrichtung Künstliche Intelligenz)

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
	180h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5h Vorlesung b) 2 SWS / 22,5h Seminar/Übung	Kontaktzeit 4 SWS / 45h h	Selbststudium 135 h	geplante Gruppengröße b) 20 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die Grundlagen des Natural Language Processing (NLP) und die wichtigsten Anwendungsgebiete. Sie können mit großen Textkörpern umgehen, geeignete NLP-Tools auswählen und diese einsetzen. Sie sind in der Lage, NLP-Modelle zu erstellen und deren Resultate quantitativ zu analysieren, zu bewerten und zu interpretieren. Die Studierenden reflektieren die Mehrdeutigkeit und Ungenauigkeit von Text und haben ein Bewusstsein für die Grenzen der Möglichkeiten aktueller NLP-Verfahren (etwa beim Erkennen von Ironie oder rhetorischen Fragen).				
3	Inhalte Moderne Computersysteme verarbeiten zunehmend Daten in natürlicher Sprache, etwa <ul style="list-style-type: none">• bei der Suche nach Texten im Internet,• in Dialogsysteme mit virtuellen Agenten,• bei der automatischen Übersetzung oder• bei der Informationsextraktion aus Nachrichtenmeldungen oder Beiträgen in sozialen Medien. In diesem Modul werden anhand von praktischen Beispielen ausgewählte aktuelle Themen des Natural Language Processing behandelt, insbesondere aus den Bereichen: <ul style="list-style-type: none">• Textklassifikation,• Erkennung von Entitäten,• Chatbots,• Topic Modelling und• Word Embeddings				
4	Lehr- und Lernformen Vorlesung und Seminar / Übung				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Machine Learning, Deep Learning, Modul Skriptsprachen (Python-Kenntnisse)				
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl: Kurdi, Mohamed; <i>Natural language processing and computational linguistics</i> , Wiley Goldberg, Yoa; <i>Neural Network Methods in Natural Language Processing</i> , Morgan & Claypool Bengfort, Benjamin; <i>Applied Text Analysis with Python</i> , O'Reilly

Operations Research

Kennnummer		Workload 180 h	Credits 6 CP	Studiensemester 4./ 6. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sollen in die Lage versetzt werden, konkrete Problemstellungen des Operations Research selbständig mathematisch modellieren und mit Hilfe der erlernten Methoden (z.B. Simplex-Verfahren) lösen zu können. Dabei soll auch der Umgang mit einem Tabellenkalkulationsprogramm (z.B. Microsoft Excel) geübt werden.					
3	Inhalte In der einsemestrigen Lehrveranstaltung Operations Research werden wichtige Verfahren und Techniken der Unternehmensforschung erläutert und an Hand von Beispielen dargestellt. Es werden insbesondere mathematische Methoden zur Lösung von Produktionsplanungs-, Transport- und Zuordnungsproblemen behandelt. Der Schwerpunkt der Vorlesung liegt in der Besprechung von Verfahren zur Lösung linearer Optimierungsprobleme (z.B. der Varianten des Simplex-Verfahrens, Transporttableau). Es werden zahlreiche konkrete Problemstellungen behandelt, die zum Teil auch mit Hilfe eines Tabellenkalkulationsprogramms (z.B. Microsoft Excel) gelöst werden. Einige der benötigten Grundlagen aus dem Bereich der Mathematik (insbesondere die Lösung linearer Gleichungssysteme) werden zu Beginn der Lehrveranstaltung wiederholt. Die Inhalte im Einzelnen sind: <ul style="list-style-type: none">• Aufgaben des Operations Research• Mathematische Grundlagen• Lineare Optimierungsprobleme<ul style="list-style-type: none">○ Graphische Lösung○ Die Varianten des Simplex-Verfahrens• Parametrische lineare Optimierung• Transportprobleme					
4	Lehr- und Lernformen Vorlesung, Übung					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Module Mathematik 1 und 2					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. rer. nat. Andreas Koop, Prof. Dr. rer. nat. Hardy Moock					
12	Sonstige Informationen					

Ökosysteme

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 4. - 6. Sem.	Häufigkeit V jedes WiSe P jedes SoSe	Dauer 2 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,25h Seminar b) 2 SWS / 22,25h Praktikum	Kontaktzeit 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learningoutcomes)/Kompetenzen Die Studierenden wissen, wie die Anwendung des Nachhaltigkeitsziels 15 lokal und global terrestrische Ökosysteme erhält. Sie können in der Ökologie Forschungsfragen entwickeln und auf die wissenschaftliche Literaturbearbeitung anwenden. Sie können natürlich und anthropogen geprägte Umwelten beschreiben. Sie kennen die wichtigsten Prinzipien von Interaktionen in terrestrischen Ökosystemen. Sie haben praktische Erfahrungen der Analyse von Umweltbelastungen auf Ökosysteme. Sie kennen die Umsetzung des Nachhaltigkeitsziels 15 der Vereinten Nationen und können die Leistungsfähigkeit von Ökosystemen beschreiben.				
3	Inhalte Grundlagen: Systembegriffe, System-Umwelt-Unterscheidung, duale Hierarchien, Individuation, Expression, Disparation, angewandte Ökologie, Nachhaltigkeitsziele der UN, Frameworks zur Literaturrecherche, Literaturverwaltungsprogramme, H5P-Editoren Organismen und Populationen: Arten, Umwelt der Organismen, Fläche und Areal, zeitliche Aspekte, Ökologische Nische, Populationsgleichung, Populationsdynamik, Evolution von Lebenszyklen, Dichteregulation und Populationsschwankungen, Systeme von Populationen Wechselwirkungen, Lebensgemeinschaften und Ökosysteme: Nahrungserwerb, trophische Ebenen, Prinzipien der Wechselwirkungen, hierarchische Beziehungen, Mutualismus, Struktur von Lebensgemeinschaften, ökologische Prozesse, Dynamik von Lebensgemeinschaften, Biogeographie, Energie-, Stoff- und Informationsfluss, Großlebensräume Nachhaltige Entwicklung an Land: Forschungsfragen gemäß LINER-Framework, Erkundung terrestrischer Ökosysteme, Literaturrecherche gemäß SPIDER-Framework, Anwendung des Global Indicator Framework, Entwicklung nachhaltiger Handlungsoptionen, Erstellung von Open Source Lernressourcen Auswahl Gruppenpraktika: <ul style="list-style-type: none"> Lokale Projekte: Nachhaltige Nutzung und Bewirtschaftung, Verlust biologischer Vielfalt bewerten und stoppen, Gerechter Zugang zu genetischen Ressourcen, Wilderei bekämpfen, Auswirkungen gebietsfremder Arten verringern Globale Projekte: Erhaltung der Bergökosysteme, Finanzielle Bedingungen für nachhaltiges Leben an Land fördern, Wüstenbildung und Bodenzerstörung bekämpfen 				
4	Lehrformen Seminar, Praktika				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				

6	Prüfungsformen Portfolioprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)
7	Prüfungsvorleistung keine
8	Voraussetzungen für die Vergabe von Kreditpunkten Portfolio erfolgreich bestanden
9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote: $6/180 = 3,33 \%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. rer. nat. Kilian Hennes, Prof. Dr. rer. nat. Eva Eisenbarth, Prof. Dr. rer. nat. Eckhard Rikowski
12	Sonstige Informationen Literaturauswahl: Wolfgang Nentwig, Sven Bacher, Roland Brandl „Ökologie kompakt“, SpringerSpektrum 2017

Partizipatives Design

Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen 4 SWS / 22,5 h Vorlesung / seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Partizipative Ansätze beziehen die künftigen Nutzer*innen von vorneherein und möglichst gleichberechtigt in den Entwicklungsprozess ein. Das Modul bereitet die Studierenden auf die daraus resultierenden Besonderheiten und Anforderungen an das Vorgehen vor. Die Studierenden lernen den Ansatz des Participatory Design bei der Entwicklung von IT kennen. Sie verfügen über Grundlagenwissen zu den Methoden und Modellen, kennen beispielhaft nach diesem Vorgehensmodell durchgeführte IT-Entwicklungsprojekte und können geeignete Werkzeuge und Techniken des Participatory Design für eigene Entwicklungsprojekte auswählen und selbstständig anwenden.				
3	Inhalte <ul style="list-style-type: none">Grundlagen der Partizipation: Beteiligte und ihre Kulturen, Grade und Formen der Partizipation, Partizipatorische Designperspektiven, Wissensdomänen, Entscheidungsfindung im PDMethoden und Prozessmodelle des PD: Software Technology for Evolutionary Participatory Systems Development (STEPS), Methods of initial analysis and design (MUST), Cooperative Experimental System Development (CESD)Werkzeuge und Techniken des PD: Tell-Make-Enact, Informationsgewinnung / -austausch, wechselseitiges Lernen, Gestaltung partizipatorische Prototypen /Mook-ups, Vorführung von ZukunftsszenarienFallbeispieleMessung und Bewertung von ErgebnisqualitätEntwurf und Durchführung eines exemplarischen Entwurfs nach PD				
4	Lehr- und Lernformen Vorlesung / seminaristischer Unterricht (50%), Praktikum (50%)				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Hausarbeit mit Fachvortrag (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Simonsen et al. (eds.), Routledge International Handbook of Participatory Design, Routledge, 2013. Bodker et al. (eds.), Participatory IT Design: Designing for Business and Workplace Realities, MIT Press, 2009. Conference PDC: Participatory Design / Proceedings of the Conference on Participatory Design, biennial (2018, 2020, 2022, ...), https://dl.acm.org/conference/pdc/proceedings Weidekamp-Maicher, Menschen mit Demenz in der partizipativen Entwicklung von Technik, Springer 2021.

Praktische Anwendungen von Algorithmen

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 3. / 5.	Häufigkeit des Angebots Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen 4 SWS / 45 h Seminar / Blockveranstaltung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen Strategien zur Lösung komplexer algorithmischer Probleme wie dynamische Programmierung oder die Greedy-Heuristik und können sie in der Praxis anwenden. Basis sind dabei konkrete Problemstellungen aus Wettbewerben wie etwa Google Hashcode.				
3	Inhalte Im Seminar erarbeiten die Studierenden in Dreier- oder Vierergruppen Lösungen zu Problemen aus vergangenen Wettbewerben zum Lösen algorithmischer Probleme, etwa Google Hashcode, und stellen ihre Lösungen und die verwendeten Strategien den anderen Studierenden vor. In ihren Gruppen nehmen die Studierenden an einem Wettbewerb wie Google Hashcode teil. Der konkrete Wettbewerb und der Termin wird den Studierenden zu Beginn des Moduls mitgeteilt.				
4	Lehrformen Seminar				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Grundlagen der Informatik				
6	Prüfungsformen Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistungen keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Im Verbundstudiengang Angewandte Informatik				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl: Baka, Benjamin <i>Python Data Structures and Algorithms</i> David Esparza Alba: <i>Algorithms: For Competitive Programming</i>

Praktische Betriebssysteme 1

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Kennennummer						Workload	Credits	Studiensemester	Häufigkeit	Dauer	
						180 h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester	
1	Lehrveranstaltungen						Kontaktzeit		Selbststudium	geplante Gruppengröße	
	a) 2 SWS / 22,5 h Vorlesung						4 SWS / 45 h		135 h	b) 15 Studierende	
	b) 2 SWS / 22, 5 h Praktikum										
2	Lernergebnisse (learning outcomes) / Kompetenzen										
	Dieses Modul vermittelt den Studierenden die Grundlagen und die nötigen Fähigkeiten und Kenntnisse zur Auswahl, Bereitstellung und Verwaltung einer Microsoft-Netzwerklösung für kleine und mittelständische sowie große Unternehmen.										
	Am Ende dieses Moduls sind die Studierenden in der Lage, folgendes auszuführen:										
	<ul style="list-style-type: none">• Entwerfen einer Netzwerklösung für kleine, mittelständische und große Unternehmen• Auswahl geeigneter Microsoft-Server• Installieren oder Aktualisieren von Microsoft-Servern• Konfigurieren von Microsoft-Servern als Datei- und Druck-Server• Konfigurieren von Microsoft-Servern als Domaincontroller• Verwalten und Überwachen von Microsoft-Servern• Absichern eines Netzwerks mit Microsoft-Servern• Verwalten des Messaging										
3	Inhalte										
	<ul style="list-style-type: none">• Umgang mit der MMC• Benutzerverwaltung, Computerverwaltung• DHCP, DNS• ADS, Gruppenrichtlinien• Installation von Microsoft-Servern• Internet- und Remote-Verbindungen• Lizenzverwaltung• Verwalten von Client-Computern, Benutzergruppen und Remote-Arbeitsplätzen• Datei- und Druckserver• Domaincontroller• Serversicherheit: Internetzugriff mittels NAT, Remotezugriff, VPN, Absichern von Datei-, Ordner- und Druckobjekten, Virenschutzmaßnahmen, Schutz vor Datenverlusten• Domain, Forest• Design einer Messaging-Infrastruktur• Verwalten und Konfigurieren von Exchange-Servern										
4	Lehr- und Lernformen										
	Vorlesung mit begleitenden Praktika (in Kleingruppen); die vorgestellten Konzepte werden in der Laborumgebung praktisch umgesetzt und in Betrieb genommen.										
5	Teilnahmevoraussetzungen										
	Formal:										
	Inhaltlich: Mathematik 1, Mathematik 2, Rechnerarchitektur, Grundlagen der Betriebssysteme, Rechnernetze										
6	Prüfungsformen										
	Klausur oder Portfolio oder praktische Prüfung										
	(Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)										
7	Prüfungsvorleistung										
	keine										

8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen

Praktische Betriebssysteme 2

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Wahlmodul der Studienschichtung Systemintegration					
Kennnummer	Workload	Credits	Studiensemester	Häufigkeit	Dauer
	180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erwerben praktische Kenntnisse in der Shell-Programmierung (Bash unter Linux) und in der Linux-Systemverwaltung. Sie verinnerlichen die Unix-Philosophie, sich für Problemlösungen des „Unix-Werkzeugkastens“ zu bedienen, also aus einer großen Sammlung einfacher Tools eine geeignete Auswahl zu treffen und diese Tools dann zu einer komplexen Problemlösung zusammenzufügen.				
3	Inhalte <ul style="list-style-type: none">• Kommandozeilenwerkzeuge, Pipes, Ein- und Ausgabe-Umleitung• Shell-Programmierung (Bash): Gängige Kontrollstrukturen (Schleifen, Fallunterscheidungen, Funktionen)• Reguläre Ausdrücke (praktischer Einsatz in Unix-Tools)• Prozesse, Prozess-Management• Automatisierung mit Cron• Dateisysteme, Dateirechte				
4	Lehr- und Lernformen Vorlesung (teilweise als Flipped Classroom) mit begleitenden Praktika (in Kleingruppen)				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur oder Portfolio oder praktische Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				
10	Stellenwert der Note für die Endnote 6/180 = 3,33%				
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer				
12	Sonstige Informationen Literaturauswahl: Ehses, Köhler, Riemer, Stenzel u. Victor, <i>Systemprogrammierung in UNIX/Linux</i> Friedl, <i>Reguläre Ausdrücke</i> Wolf, <i>Shell-Programmierung: Das umfassende Handbuch</i>				

Programmierung graphischer Benutzeroberflächen mit Java

(Wahlpflichtmodul der Studienrichtung Anwendungsentwicklung)

Wahlprüfungsausschuss der Studiengänge Anwendungsentwicklung/						
Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erhalten die erforderlichen Kenntnisse über die Erstellung von graphischen Benutzeroberflächen der JavaFX-Bibliotheken. Sie kennen darüber hinaus erprobte Utility-Klassen und Frameworks für diesen Problembereich.					
3	Inhalte Der Inhalt der Vorlesung umfasst die Themen <ul style="list-style-type: none">• Grundlegende Konzepte von JavaFX• Dynamische Layouts• JavaFX-Komponenten im Überblick• Utility-Klassen und Frameworks zur Steuerung graphischer Oberflächen Im Praktikum werden auf die Vorlesung abgestimmte Präsenzaufgaben bearbeitet.					
4	Lehr- und Lernformen Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Objektorientierte Programmierung					
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					
10	Stellenwert der Note für die Endnote 6/180 = 3,33%					
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Andreas Steins					
12	Sonstige Informationen Literaturauswahl: Client Technologies: Java Platform, Standard Edition (Java SE) 8 Release 8, http://docs.oracle.com/javase/8/javase-clienttechnologies.htm C. Dea, M. Heckler et al, JavaFX 8 - Introduction by Example, Apress K. Sharan, Learn JavaFX 8 – Building User Experience and Interfaces with Java 8, Apress J. Vos, W. Gao et al, Pro JavaFX 8 – A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients, Apress					

Quantencomputing

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 4-6. Sem.	Häufigkeit Sommer- und Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die mathematischen und physikalischen Grundlagen des Quantencomputing. Sie kennen und können Quantenschaltkreise entwerfen und in anwendungsorientierte Szenarien anwenden.				
3	Inhalte <ul style="list-style-type: none"> • Spezielle mathematische Grundlagen des Quantencomputing • Quantenmechanische Prinzipien • Vom Quantenbit zum Quantenschaltkreise • Quantenlogik • Quantencomputer – Stand der Technik • Entwicklungsumgebung für Quantencomputer am Beispiel Qiskit • Anwendungen des Quantencomputing u.a.: <ul style="list-style-type: none"> ○ Quanten-Fouriertransformation ○ Quanten-Suchalgorithmen ○ Quantenkommunikation ○ Shore Faktorisierungs-Algorithmus ○ Teleportation ○ Quantenkryptographie 				
	Lehr- und Lernformen Vorlesungen mit begleitenden Praktika in Kleingruppen (< 15 Studierende); Die vorgestellten Verfahren werden an Beispielen praktisch angewandt und vertieft.				
5	Teilnahmevoraussetzungen Formal: „Grundlagen der Informatik 1 und 2“; „Mathematik 1 und 2“ Inhaltlich: „Grundlagen der Programmierung“, „Grundlagen der Informatik 1 und 2“, „Mathematik 1 und 2“				
6	Prüfungsformen Klausur oder eine andere Prüfungsform der RPO oder FPO (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,3%
11	Modulbeauftragter und hauptamtlich Lehrende Prof. Dr. Michael Rübsam
12	Sonstige Informationen Literaturauswahl: Matthias Homeister, Quanten Computing verstehen. Springer Vieweg, 6. Auflage, 2022 Andreas de Vries, Quantenrechnen – Eine Einführung in Quantum Computation für Ingenieure und Informatiker. Books on Demand, 2012 Bettina Just, Quantencomputing kompakt - Spukhafte Fernwirkung und Teleportation endlich verständlich. Springer Vieweg, 2020 Joachim Stolze, Dieter Suter, Quantum Computing – A Short Course from Theory to Experiment. Wiley-VCH, 2008

Rechnernetze 2

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Wahlmodul der Studiengrichtung Systemintegration					
Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 4. / 6. Semester	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen In dieser Lehrveranstaltung werden aufbauend auf dem Grundlagenmodul Rechnernetze vertiefende Kenntnisse über hoch verfügbare, echtzeitfähige Netzinfrastrukturen vermittelt. Die erworbenen theoretischen Kenntnisse über moderne Netzinfrastrukturen sowie die Fähigkeiten zur Bedarfsanalyse, Planung, Konfiguration, Fehlersuche und zum Betrieb der Netzinfrastrukturen werden in umfangreichen Laborübungen angewandt und vertieft. Im Rahmen dieser Lehrveranstaltung können die Studierenden an den beiden CCNA-Kursen „Networking Fundamentals“ und „Routing Protocols and Concepts“ teilnehmen, die auf die industriezertifizierte CCNA (Cisco Certified Network Associate) Prüfung vorbereiten.				
3	Inhalte <ul style="list-style-type: none">• Verfügbarkeit und Redundanz• Designprinzipien für hoch verfügbare echtzeitfähige Netzwerkinfrastrukturen• VLAN-Konzepte• Spanning Tree• IPv4 und IPv6 Adressen (u.a. VLSM, CIDR, supernetting)• Interior Gateway Routing-Protokolle<ol style="list-style-type: none">1. Distance Vector Protokolle am Beispiel von RIPv1 und RIPv22. Link State Protokolle am Beispiel von OSPF• Exterior Gateway Routing-Protokolle<ol style="list-style-type: none">3. Path Vector Protokolle am Beispiel von BGP• Quality of Service• Multicast: IGMP, Multicastingprotokolle				
4	Lehr- und Lernformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Rechnernetze 1				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Praktikum – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam
12	Sonstige Informationen

Rechnungswesen 1

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 4. / 6. Sem.	Häufigkeit Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden sollen in der Lage sein, die einzelnen Elemente des Jahresabschlusses zu beschreiben und wichtige Informationen aus dem Jahresabschluss zur Beurteilung der Unternehmenssituation zu entnehmen. Im Bereich der Kostenrechnung sollen die Studierenden grundlegende Begriffe kennen, zwischen verschiedenen Kostenrechnungssystemen unterscheiden können und die Ist- Kostenrechnung auf Vollkostenbasis beherrschen. Detaillierte Lernziele werden im Rahmen der Veranstaltung bekannt gegeben.</p>				
3	Inhalte 1. Überblick <ul style="list-style-type: none"> • Begriff und Aufgaben des Rechnungswesens • Teilgebiete des Rechnungswesens 2. Externes Rechnungswesen (Jahresabschluss) <ul style="list-style-type: none"> • Inventur/Inventar/Bilanz • Gewinn- und Verlustrechnung (GuV) • Anhang und Lagebericht 3. Internes Rechnungswesen (Kostenrechnung) 3.1. Grundlagen <ul style="list-style-type: none"> • Aufgaben und Grundbegriffe • Systeme der Kostenrechnung 3.2. Ist-Kostenrechnung auf Vollkostenbasis <ul style="list-style-type: none"> • Kostenartenrechnung • Kostenstellenrechnung • Kostenträgerstückrechnung (Kalkulation) • Kostenträgerzeitrechnung 				
4	Lehr- und Lernformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Übung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) In Studiengängen des Nachbarfachbereichs Maschinenbau
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Lothar Winnen
12	Sonstige Informationen Literaturauswahl: Haberstock, L.: Kostenrechnung I, Berlin Kloock, J./Sieben, G./Schildbach, Th.: Kosten- und Leistungsrechnung, Düsseldorf Schierenbeck, H.: Grundzüge der Betriebswirtschaftslehre, München/Wien Schweitzer, M./Hettich, O./Küpper, H.-U.: Systeme der Kosten- und Leistungsrechnung, München Thommen, J.-P./Achleitner, A.-K.: Allgemeine Betriebswirtschaftslehre. Umfassende Einführung aus managementorientierter Sicht, Wiesbaden Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre, München

Rechnungswesen 2

Kennnummer	Workload 180h	Credits 6 CP	Studiensemester 3. / 5. Semester	Häufigkeit Wintersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 25 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Zunächst sollen die Studierenden weitere Kostenarten und Möglichkeiten zur Verrechnung innerbetrieblicher Leistungen kennen lernen. Des Weiteren sollen die Studierenden erfahren, dass nur mit Hilfe neuerer Kostenrechnungsverfahren optimale Wirtschaftlichkeitskontrollen möglich sind sowie entscheidungsrelevante Informationen zur effizienten Gestaltung des Unternehmens zur Verfügung gestellt werden können. Deshalb sollen die Studierenden die Plankosten, die Deckungsbeitrags- und die Prozesskostenrechnung sowie das Target Costing kennen und anwenden lernen können. Detaillierte Lernziele werden im Rahmen der Veranstaltung bekannt gegeben.				
3	Inhalte <ul style="list-style-type: none"> • Vertiefung der Istkostenrechnung auf Vollkostenbasis <ul style="list-style-type: none"> ○ weitere Kostenarten ○ weitere Verfahren der innerbetrieblichen Leistungsverrechnung • Deckungsbeitragsrechnung <ul style="list-style-type: none"> ○ Programmplanung ohne und mit Engpässen ○ Eigenfertigung und Fremdbezug • Wahl des optimalen Produktionsverfahrens • Plankostenrechnung <ul style="list-style-type: none"> ○ starre Plankostenrechnung ○ flexible Plankostenrechnung auf Voll- und Teilkostenbasis • Neue Instrumente des Kostenmanagements <ul style="list-style-type: none"> ○ Prozesskostenmanagement ○ Target Costing 				
4	Lehr- und Lernformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Rechnungswesen 1				
6	Prüfungsformen Klausur (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				
7	Prüfungsvorleistung Studienleistung für Übung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) keine				

10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Lothar Winnen
12	Sonstige Informationen Literaturauswahl: Haberstock, L.: Kostenrechnung I, Berlin Haberstock, L.: Kostenrechnung II, Berlin Kilger, W.: Flexible Plankostenrechnung und Deckungsbeitragsrechnung, Wiesbaden Kloock, J./Sieben, G./Schildbach, Th.: Kosten- und Leistungsrechnung, Düsseldorf Schierenbeck, H.: Grundzüge der Betriebswirtschaftslehre, München/Wien Schweitzer, M./Hettich, O./Küpper, H.-U.: Systeme der Kostenrechnung- und Leistungsrechnung, München Thommen, J.-P./Achleitner, A.-K.: Allgemeine Betriebswirtschaftslehre. Umfassende Einführung aus managementorientierter Sicht Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre, München

Statistik

(Wahlpflichtmodul der Studienrichtung Künstliche Intelligenz)

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit des Angebots	Dauer
		180 h	6 CP	4. / 6. Sem.	Sommersemester	1 Semester
1	Lehrveranstaltungen a) 4 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Übung		Kontaktzeit 4 SWS / 45 h		Selbststudium 90 h	geplante Gruppengröße b) 30 Studierende
2	Lernergebnisse (learningoutcomes) / Kompetenzen Als fachliche Qualifikation lernen die Studierenden die behandelten statistischen Methoden anzuwenden. Sie gewinnen damit Informationen aus Datenmaterial und können dieses hinsichtlich unterschiedlicher Fragestellungen auswerten. Sie ziehen Schlussfolgerungen aus der Hypothesenüberprüfung, und erlangen als überfachliche Qualifikation die Fähigkeit, Entscheidungen unter ungewissen Bedingungen vorbereiten und technische Prozesse auf ihre Tauglichkeit überprüfen. Sie können die aus statistischen Untersuchungen gewonnenen Ergebnisse darstellen und hinsichtlich Korrektheit sowie Aussagekraft beurteilen. Die wissenschaftlich korrekte Anwendung statistischer Methoden stellt eine Selbstkompetenz und ein überfachliches Qualifikationsziel dar, weil sie eine universelle Bedeutung hat zur Beurteilung verschiedenen Datenmaterials.					
3	Inhalte <ul style="list-style-type: none">• Deskriptive Statistik Statistische Einheit, Grundgesamtheit und Stichproben, Absolute und relative Häufigkeit, Graphische Darstellungsmöglichkeiten von Häufigkeiten, Maßzahlen in der Häufigkeitsverteilung z.B. Mittelwerte, Streuungsmaße, Korrelationskoeffizient• Kombinatorik• Grundbegriffe der Wahrscheinlichkeitsrechnung Ereignisse und Wahrscheinlichkeiten, Laplace Experiment, Bedingte Wahrscheinlichkeiten und der Satz von Bayes, Anwendungsbeispiele• Diskrete Wahrscheinlichkeitsverteilungen Binomialverteilung, Erwartungswert und Varianz, Anwendungsbeispiele• Stetige Wahrscheinlichkeitsverteilungen Dichte- und Verteilungsfunktion, Erwartungswert und Varianz, Normalverteilung, Chi-Quadrat- Verteilung, Anwendungsbeispiele• Analytische Statistik Schätzen von Parametern, Testen von Hypothesen, Anwendungsbeispiele					
4	Lehr- und Lernformen Vorlesung und Übungen					
5	Teilnahmevoraussetzungen Formal: Inhaltlich: Grundlagen der Differential- und Integralrechnung					
6	Prüfungsformen Klausur 90 Minuten (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					

9	Verwendung des Moduls (in anderen Studiengängen) im Studiengang Life Science Analytics (BSc.)
10	Stellenwert der Note für die Endnote $6/180 = 3,33 \%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. rer. nat. Michael Rübsam
12	Sonstige Informationen Literaturauswahl: Monika Reimpell: Studienbuch Statistik, Wissenschaftliche Genossenschaft Südwestfalen

Systembiologie

Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180 h	6 CP	3. / 5. Sem.	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5h Vorlesung b) 2 SWS / 22,5h Praktikum		Kontaktzeit 4 SWS / 45 h		Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende
2	Lernergebnisse (learningoutcomes) / Kompetenzen Die Studierenden kennen die begrifflichen Grundlagen der Systembiologie und verstehen grundlegende Prinzipien der Übertragung von Phänomenen lebendiger Systeme in mathematische Modelle. Sie haben eine Vorstellung von den Möglichkeiten und Grenzen dieser wissenschaftlichen Strategie. Die Studierenden sind in der Lage, die Grundkonzepte der Signalverarbeitung in Lebewesen zu erklären. Sie können den Begriff Omics einordnen und für verschiedene biologische Regelsysteme anwenden. Sie können dynamische Systeme und deren Eigenschaften analysieren. Sie kennen grundlegende Methoden der Systembiologie / bioinformatischen Algorithmen, Sie kennen die Möglichkeiten cloudbasierter Ansätze der Systembiologie Sie kennen grundlegende mathematische Modelle biologischer Systeme und können mit ihnen konkrete Systeme modellieren. Sie üben das Arbeiten in interdisziplinären Teams Die Studierenden vertiefen die Vorlesungsinhalte mit praktischen Übungen zum Heranziehen einfacher mikrobieller Systeme und modellieren das Wachstum durch geeignete Algorithmen					
3	Inhalte Praktikum: Zelluläre Automaten (2D) als Game of Life, Simulation des Wachstums von Bakterienkulturen bei Variation von 2 und 3 Parametern modellieren und mit lebender Kultur abgleichen unter Nutzung automatischer Zellzähler / Photometer.					
4	Lehr- und Lernformen Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen Formal: Inhaltlich:					
6	Prüfungsformen Portfolio (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung keine					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen)					

10	Stellenwert der Note für die Endnote 6/180 = 3,33 %
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron, Prof. Dr. Eva Eisenbarth
12	Sonstige Informationen Literaturauswahl: Eberhard Voit: <i>A first Course in Systems Biology</i> , Garland Science Marian Walhout, Marc Vidal, Job Dekker: <i>Handbook of Systems Biology: Concepts and Insights</i> - (Englisch) Gebundene Ausgabe, 15. November 2012 Edda Klipp, Wolfram Liebermeister, Christoph Wierling, Axel Kowald; <i>Systems Biology: A Textbook</i> - Englisch) Taschenbuch, 20. April 2016 Yoram Vodovotz and Gary: <i>An Translational Systems Biology, Concepts and Practice for the Future of Biomedical Rese</i>

Umweltinformationssysteme (UIS)

Kennnummer	Workload 180 h	Credits 6 CP	Studiensemester 4. / 6. Sem.	Häufigkeit Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen a) 1 SWS / 11,25 h Vorlesung b) 3 SWS / 33,75 h seminaristischer Unterricht	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learningoutcomes)/Kompetenzen <ul style="list-style-type: none">Die Studierenden kennen die grundlegenden Aufgaben, Konzepte und Methoden umweltbezogener Informationsverarbeitung sowie die Architektur und Services von Umwelt- und Geoinformationssystemen.Sie kennen die Standards und Verfahren für die Erfassung, Aufbereitung, Analyse und Visualisierung von Umweltdaten und die damit zusammenhängenden Problemstellungen für UIS-Anwendungen.Sie können entsprechende Werkzeuge zur Entwicklung von UIS-Anwendungen praktisch einsetzen und selbständig ein einfaches UIS-Projekt planen sowie dieses auf Basis freier (bzw. ggf. kommerzieller) Geoinformationssysteme umsetzen.Sie sind in der Lage in Projekten interdisziplinär und lösungsorientiert zusammen zu arbeiten mit Umweltwissenschaftlern, Biologen, Ingenieuren einerseits und Management/Verwaltung/ Informationsnutzenden andererseits.				
3	Inhalte <ul style="list-style-type: none">Einführung in UIS<ul style="list-style-type: none">Aufgaben und Ziele umweltbezogener InformationsverarbeitungAnwendungsbereiche für UIS (int./Bund/Land/Kommune, betrieblich, NGO), Berichts-/Management-/Entscheidungsunterstützungssystemegesetzlicher Rahmen, Anforderungen/Bewertungskriterien, Evaluation von UISGrundlagen Umwelt- und Geoinformation<ul style="list-style-type: none">Standards, Metadaten, Geodaten & Bezugssysteme, Datenformate/-strukturen, DatenqualitätErfassung von Umweltdaten (Umweltmessnetze, Laboranalytik, Bioindikation, Fernerkundung)Datenaufbereitung, Datenanalyse (Aggregation, Klassifikation, Umweltdatenstatistik/Geostatistik, Umwelt-Data-Mining)Visualisierung raum- und umweltbezogener InformationenUIS-Anwendungen umsetzen<ul style="list-style-type: none">Architektur und Funktionen von Umweltinformationssystemen und GIS, Umwelt-/Geodatendienste und ServicesWerkzeuge und Produkte, freie GIS-Systeme (z.B. QGIS), kommerzielle GIS-Systeme (z.B. ArcGIS)Vorstellung und Analyse ausgewählter Praxisbeispiele zu UIS-Projekten und -AnwendungenNutzung von Open Data, Umweltportalen und UmweltdatenkatalogenSemesterbegleitend Projekt zu UIS, vorzugsweise mit externen Partnern<ul style="list-style-type: none">Projektauswahl/-definitionAnleitung und Begleitung der Teams bei Projektplanung und Durchführung				
4	Lehrformen Vorlesung, seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: Inhaltlich:				
6	Prüfungsformen Prozessorientierte Prüfungsleistung, Portfolio oder Kombinationsprüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				

7	Prüfungsvorleistung keine
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen)
10	Stellenwert der Note für die Endnote: $6/180 = 3,33 \%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. RyLee Hühne
12	Sonstige Informationen Literaturauswahl: Freitag et al. (Hrsg.) Umweltinformationssysteme, Springer Verlag Fischer-Stabel, Umweltinformationssysteme, Wichmann, Neuauflage im Druck de Lange, Geoinformatik in Theorie und Praxis, Springer Verlag, 2020 Information Resources Management Association (Hrsg.), Environmental Information Systems: Concepts, Methodologies, Tools, and Applications, IGI Global QGIS project, QGIS Übungshandbuch, online https://docs.qgis.org

Virtualisierung

(Wahlpflichtmodul der Studienrichtung Systemintegration)

Wahlprüfungsmodul der Studiengrichtung Systemintegration						
Kennnummer		Workload	Credits	Studiensemester	Häufigkeit	Dauer
		180 h	6 CP	5. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5 h Vorlesung b) 2 SWS / 22,5 h Praktikum		Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Den Studierenden sind die verschiedenen Konzepte der Virtualisierung bekannt und Sie können deren Einsatzgebiete sowie die jeweiligen Vor- und Nachteile benennen. Sie kennen die verschiedenen Systemkomponenten einer im Rechenzentrum führenden Virtualisierungslösung, können diese einsetzen, um einfache virtuelle Infrastrukturen aufzubauen und die Konzepte auf vergleichbare Frameworks anderer Hersteller übertragen. Die Studierenden kennen die Grundlagen der Betriebssystem-Virtualisierung und können die Vor- und Nachteile im Vergleich zur Hypervisor-basierten Virtualisierung benennen. Sie können Container Frameworks (z.B. Docker) einsetzen, um Container zu managen und eigene Container-Images zu erzeugen.					
3	Inhalte <ul style="list-style-type: none">• Motivation und Grundlagen der Virtualisierung• Hypervisor-Technologien• Server-Virtualisierung• Virtualisierung im Rechenzentrum• Hochverfügbarkeit und Fehlertoleranz• Hardwareunterstützung für Virtualisierung• Betriebssystem-Virtualisierung (Container)• Virtuelle Maschinen für Programmiersprachen					
4	Lehr- und Lernformen Vorlesung (50%), Praktikum (50%)					
5	Teilnahmevoraussetzungen Formal: Inhaltlich:					
6	Prüfungsformen Mündliche Prüfung (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)					
7	Prüfungsvorleistung Studienleistung für Praktikum– die genauen Modalitäten werden zu Semesterbeginn kommuniziert					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen) keine					

10	Stellenwert der Note für die Endnote 6/180 = 3,33%
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl: Jim Smith und Ravi Nair: <i>Virtual Machines: Versatile Platforms for Systems and Processes</i> . Morgan Kaufmann Publishers, 2005 Edouard Bugnion, Jason Nieh, und Dan Tsafir: <i>Hardware and Software Support for Virtualization</i> , Morgan & Claypool Publishers, 2017 Ralph Göpel: <i>Praxishandbuch VMware vSphere 6.7</i> , O'Reilly, 2019 VMware, Inc.: <i>vCenter Server und Hostverwaltung</i> , 2018 VMware, Inc.: <i>vSphere Networking</i> , 2018 VMware, Inc.: <i>Handbuch zur Verfügbarkeit in vSphere</i> , 2018 Bernd Öggl, Michael Kofler: <i>Docker – Das Praxisbuch für Entwickler und DevOps-Teams</i> , Rheinwerk, 2018

Container

Folgende Container sind diesem Studiengang zugeordnet:

- Anwendungsentwicklung
- Künstliche Intelligenz
- Systemintegration
- Querschnittsthemen

Die Module, die den einzelnen Containern zugeordnet sind, finden Sie in den nächsten Abschnitten

Container Anwendungsentwicklung

Cloud-native Software Development

Kennnummer	Workload	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
	180 h	6 CP	5. Semester	nach Bedarf im Wintersemester	1 Semester
1	Lehrveranstaltungen a) 2 SWS / 22,5h Vorlesung b) 2 SWS / 22,5h Praktikum	Kontaktzeit 4 SWS / 45h	Selbststudium 135h	geplante Gruppengröße b) 15 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen In diesem Modul erwerben die Studierenden fundierte Kenntnisse zu Technologien wie Docker, Microservices und CI/CD. Sie verstehen die Prinzipien Cloud-nativer Architekturen und deren Vorteile, die Rolle von Docker-Images für Skalierbarkeit sowie die Kommunikationsmöglichkeiten durch API-Designs und Messaging-Systeme. Darüber hinaus werden ihnen verschiedene Deployment-Strategien samt ihrer Einsatzmöglichkeiten vermittelt. Die praktische Anwendung steht im Vordergrund: Studierende entwickeln containerisierte Anwendungen, berücksichtigen Skalierbarkeit und Sicherheitsaspekte und arbeiten in Teams an Projekten. Sie analysieren systematisch die Unterschiede zwischen monolithischen und Cloud-nativen Architekturen sowie Herausforderungen der sicheren und skalierbaren Anwendungsentwicklung. Zudem entwerfen sie eigenständige Konzepte für Cloud-native Anwendungen und bewerten deren Effizienz und Sicherheit. Kritische Reflexion und Präsentation der Ergebnisse runden das Modul ab und bereiten sie darauf vor, praxisnahe Aufgabenstellungen erfolgreich zu meistern.				
3	Inhalte <ul style="list-style-type: none">• Einführung in die Cloud-native Softwareentwicklung• Container-Technologien und Orchestrierung• API-Design und -Integration• Deployment-Strategien• Cloud-native Architekturen und Messaging-Systeme• Sicherheitskonzepte und Resilienz• CI/CD und DevOps• Integration und Skalierung von Services• Praktische Projektarbeiten und Präsentationen				
4	Lehr- und Lernformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: Teilnahme ab dem 5. Fachsemester Inhaltlich: Softwaretechnische Grundkenntnisse (Java, C++, Python) sowie Umgang mit Linux Shell erforderlich				
6	Prüfungsformen Portfolio (Die Prüfungsform kann sich ggf. ändern. – Die finale Prüfungsform ist dem Prüfungsplan zu entnehmen.)				

7	Prüfungsvorleistung keine
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote $6/180 = 3,33\%$
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr.-Ing. Thomas Dorka
12	Sonstige Informationen

Container Künstliche Intelligenz

Derzeit keine zusätzlichen Module zugeordnet

Container Systemintegration

Derzeit keine zusätzlichen Module zugeordnet

Container Querschnittsthemen

Derzeit keine zusätzlichen Module zugeordnet