

Modulhandbuch

»Informatik«

Bachelor

SPO 2019



Verabschiedet am: 02.04.2025

Die Modulbeschreibungen dienen der inhaltlichen Orientierung in Ihrem Studium.

Rechtlich verbindlich ist nur die jeweils geltende Studien- und Prüfungsordnung.

Inhaltsverzeichnis

1 Informatik Bachelor - 1. Semester	4
1.1 Programmieren 1	4
1.2 Software-Engineering 1	6
1.3 Mathematik 1	8
1.4 Grundlagen der Informatik 1	10
1.5 Rechnerstrukturen 1	14
2 Informatik Bachelor - 2. Semester	18
2.1 Programmieren 2	18
2.2 Software-Engineering 2	20
2.3 Mathematik 2	22
2.4 Grundlagen der Informatik 2	26
2.5 Datenkommunikation	30
3 Informatik Bachelor - 3. Semester	32
3.1 Programmieren 3	32
3.2 Software-Engineering 3	36
3.3 Statistik	40
3.4 Datenbanken	42
4 Informatik Bachelor - 4. Semester	46
4.1 Numerische Mathematik	46
4.2 Rechnerstrukturen 2	50
4.3 Betriebssysteme	54
4.4 Projektarbeit 1	56
5 Informatik Bachelor - 5. Semester	58
5.1 Praktische Tätigkeit (Praxissemester)	58
5.2 Praxis-Seminar	60
5.3 Systemnahe Programmierung	62
6 Informatik Bachelor - 6. Semester	66
6.1 Praktikum DVA	66
6.2 Projektarbeit 2	70
6.3 Englisch	72
6.4 Betriebswirtschaftslehre und DV-Recht	74
7 Informatik Bachelor - 7. Semester	78
7.1 Bachelor-Seminar	78
7.2 Bachelorarbeit	80
8 Wahlpflichtfächer	82
8.1 Allgemeinwissenschaftliches Wahlpflichtfach	82

8.2 Fachbezogene Wahlpflichtfächer	84
--	----

1 Informatik Bachelor - 1. Semester

1.1 Programmieren 1

Name / engl.

Programmieren 1 / Programming 1

Kürzel

PRG1

Verantwortlicher

Prof. Dr. Lothar Braun

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 8,

Präsenzzeit: 90 h, Selbststudium: 150 h, Gesamtaufwand: 240 h

Lehrveranstaltungen

Programmieren 1 (4 SWS)

Praktikum Programmieren 1 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970010

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Programmieren 1

Prüfungsform

Elektronische Prüfung, 60 Minuten, keine Hilfsmittel

Inhalte des Moduls

Systematische Einführung in Syntax, Semantik und Pragmatik einer zeitgemäßen objektorientierten Sprache (Java) in Verbindung mit der Anwendung von objektorientierten Prinzipien.

Die wichtigsten Themenbereiche:

- Typsystem
- Kontrollstrukturen
- Objekte und Klassen
- Methoden und Attribute, Kapselung
- Vererbung und Polymorphismus
- Fehlerbehandlung
- Ressourcen: Laufzeit, Speicher und dessen Verwaltung
- Tools: Compiler, Interpreter, Debugger, IDE

Die in der Vorlesung präsentierten Inhalte werden in einem begleitenden Praktikum an diversen Aufgaben eingeübt und vertieft.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- beherrschen die Grundkonzepte einer objektorientierten Programmiersprache.
- erlangen die praktische Fähigkeit, diese im Rahmen von kleineren Problemstellungen selbständig anwenden zu können.

Literaturliste

Aus der Vielzahl von Java-Büchern als Begleitung zur Vorlesung besonders geeignet:

Goll. J. al: Java als erste Programmiersprache. Teubner.

Ullenboom Christian: Java ist auch eine Insel. Galileo Computing. Online Ausgabe:
<http://www.tutego.com/javabuch/online.htm>

1.2 Software-Engineering 1

Name / engl.

Software-Engineering 1 / Software-Engineering 1

Kürzel

SE1

Verantwortlicher

Prof. Dr. Anja Metzner

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Software-Engineering 1 (2 SWS)

Praktikum Software-Engineering 1 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970030

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Software-Engineering 1

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Inhalte des Moduls

Diese Vorlesung führt in die Grundlagen des Software Engineerings ein. Software Engineering umfasst vielfältige Verfahren zur Komplexitätsbewältigung des gesamten Erstellungs- und Lebenszykluses eines Software-Systems.

- Einführung in Software Engineering
- Der Lebenszyklus von Software Systemen
- Vorgehensmodelle
- Planungsphase
- Definitionsphase und Requirements Engineering
- Software Designphase
- Verifikation und Validation: Testen von Software
- Software Wartung

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- wichtige Fachbegriffe des Software Engineering zu beschreiben
- die Bedeutung und die Notwendigkeit von Software Engineering zu verstehen
- Ablauf und Aufgaben des Software Engineerings zu verstehen
- Erste, ausgewählte Software Engineering Methoden anzuwenden
- Wesentliche UML-Diagramme zu verstehen und selbst zu entwickeln
- Software Architekturen zu erkennen

Literaturliste

Begleitend zur Vorlesung:

Anja Metzner: Software Engineering - kompakt, 1. Auflage, Carl Hanser Verlag, 2020

Bernd Österreich, Axel Scheithauer: Die UML-Kurzreferenz 2.5 für die Praxis: kurz, bündig, ballastfrei, 6. Auflage, Oldenbourg Wissenschaftsverlag, 2014

Christine Rupp und SOPHISTen: Requirements-Engineering und -Management: Das Handbuch für Anforderungen in jeder Situation, 7.Auflage, Carl Hanser Verlag, 2020

Ian Sommerville: Software Engineering, 10.Auflage, Pearson Studium, Addison-Wesley, 2018

1.3 Mathematik 1

Name / engl.

Mathematik 1 / Mathematics 1

Kürzel

MAT1

Verantwortlicher

Prof. Dr. Stefan Glasauer

Lehrsprache

Deutsch

Fakultät

Fakultät für Geistes- und Naturwissenschaften

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,

Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Mathematik 1 (5 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

3970050

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 90 Minuten, keine Hilfsmittel

Inhalte des Moduls

- Logik und Mengenlehre
- Induktion und Rekursion
- Grundlagen der Analysis
- Funktionen einer Variablen
- Differenzialrechnung für Funktionen einer Variablen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Vorlesung führt in die Mathematik auf Hochschulniveau ein.

Die Studierenden

- wiederholen und vertiefen auch Inhalte aus dem Lehrplan der Fachoberschule (Ausbildungsrichtung Technik).
- erlangen sowohl Rechenfertigkeiten als auch ein grundlegendes Verständnis von mathematischen Methoden und Zusammenhängen.
- sind in der Lage, mathematische Methoden in der Informatik und ihren Anwendungsgebieten zur Problemlösung einzusetzen.
- sind ferner in der Lage, sich im Studium und Beruf benötigte mathematische Hilfsmittel anhand von Lehrbuchliteratur nach Bedarf selbst anzueignen.

Literaturliste

Teschl/Teschl: Mathematik für Informatiker 1 und 2, Springer.

Hartmann: Mathematik für Informatiker, Vieweg + Teubner.

1.4 Grundlagen der Informatik 1

Name / engl.

Grundlagen der Informatik 1 / Fundamentals of Computer Sciences 1

Kürzel

GDI1

Verantwortlicher

Prof. Dr. Markus Degen

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Grundlagen der Informatik 1 (3 SWS)

Praktikum Grundlagen der Informatik 1 (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970070

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Grundlagen der Informatik 1

Prüfungsform

Klausur, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner (kein Smartphone)

(10% Bonuspunkte durch Studienarbeit in Praktikum möglich)

Inhalte des Moduls

Das Modul GDI 1 führt in die Grundlagen der Theoretischen Informatik ein. Nach einer allgemeinen Einführung in die Teilgebiete der Informatik und der Klärung des Algorithmus-Begriffs widmet sich das Modul den Schwerpunkten

- Automatentheorie
- formale Sprachen
- Grammatiken und
- Grundlagen der Berechenbarkeit.

Das Modul legt Wert darauf zu zeigen, wie Methoden der theoretischen Informatik für Aufgabenstellungen der praktischen und technischen Informatik, wie z.B. der Entwicklung von Scannern und Parsern für formale Sprachen und Steuerungsfunktionalität für technische Systeme eingesetzt werden können.

Das Modul ist in einen Vorlesungs- und einen praktischen Teil untergliedert, in dem die Studierenden Übungsaufgaben zu den vorgestellten Inhalten lösen und Modelle und Algorithmen zu Konzepten der theoretischen Informatik entwickeln und präsentieren.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul Grundlagen der Informatik 1 sind die Studierenden in der Lage

- Algorithmen zu spezifizieren,
- die verschiedenen Automatentypen zu unterscheiden
- deren Mächtigkeit zu beurteilen
- Automaten anzuwenden, um ausgewählte Fragestellungen der lexikalischen Analyse und der Syntaxanalyse von Programmen oder Datenbeschreibungen zu lösen.

Sie können formale Automaten anwenden, um das Verhalten technischer Systeme zu modellieren und zu analysieren.

Durch die Beschäftigung mit der Theorie der Berechenbarkeit begegnen die Studierenden erstmals auch den Grenzen dessen, was Computer oder technische Systeme leisten können.

Durch die Beschäftigung mit Turing-Maschinen und äquivalenten Programmiermodellen, erlernen die Studierenden, welche Basiszutaten erforderlich sind, um alle algorithmisch beschreibbaren Probleme zu lösen.

Literaturliste

Socher, R.: Theoretische Grundlagen der Informatik, 3. Auflage, Hanser, 2008

Herold, H., Lurz, B.: Grundlagen der Informatik, 3. Auflage, Pearson, 2017

Schöning, U.: Theoretische Informatik - kurz gefasst, Springer, 2008

Hofmann, D. W.: Theoretische Informatik, Hanser, 2015

Karstens, U., Kleine Büning, H.: Modellierung: Grundlagen und Formale Methoden, Hanser, 2008

1.5 Rechnerstrukturen 1

Name / engl.

Rechnerstrukturen 1 / Computer Structures 1

Kürzel

REC1

Verantwortlicher

Prof. Dr. Volodymyr Brovkov

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,

Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Rechnerstrukturen 1 (5 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht

Prüfung

Prüfungsnummer

3970090

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner, 2 DIN-A4-Seiten handgeschriebene Formelsammlung

Inhalte des Moduls

Grundlagen:

- Informationsdarstellung
- Binärarithmetik
- Schaltnetze
- Schaltwerke

Rechnerkomponenten:

- Maschinenbefehle
- ALU
- Hauptspeicher
- Systembus
- Prozessor
- Typische Datenwege

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Darstellung von unterschiedlichen Datentypen im Rechnerspeicher zu beschreiben.
- Grundoperationen der Binärarithmetik für Ganzzahlen, Fest- und Gleitkommazahlen zu erklären und einfache numerische Beispiele zu evaluieren.
- Typische Bestandteile eines Rechners zu kennen und dessen Zusammenspiel zu erklären.
- Technische Implementierung von wesentlichen Baugruppen eines Prozessors zu kennen.
- Datenpfade bei Ausführung von typischen Maschinenbefehlen zu beschreiben und Zusammenspiel von wesentlichen Prozessorbaugruppen zu analysieren.

Literaturliste

Hoffmann, D.: Grundlagen der technischen Informatik. Carl Hanser Verlag München, 2016.

Patterson, D., Hennessy, J.: Rechnerorganisation und Rechnerentwurf. De Gruyter Studium Oldenbourg, 2016.

Fertig, A.: Rechnerarchitektur Grundlagen. BoD, Norderstedt, 2016.

Hellmann, R.: Rechnerarchitektur: Einführung in den Aufbau moderner Computer. Oldenbourg, 2013.

Malz, H.: Rechnerarchitektur. Vieweg, Braunschweig, 2004.

Märting, C.: Einführung in die Rechnerarchitektur. Hanser Verlag, München, 2003.

2 Informatik Bachelor - 2. Semester

2.1 Programmieren 2

Name / engl.

Programmieren 2 / Programming 2

Kürzel

PRG2

Verantwortlicher

Prof. Dr. Lothar Braun

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 8,

Präsenzzeit: 90 h, Selbststudium: 150 h, Gesamtaufwand: 240 h

Lehrveranstaltungen

Programmieren 2 (4 SWS)

Praktikum Programmieren 2 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht zur Vermittlung grundlegender Vorgehensweisen und inhaltlicher Zusammenhänge. Praktikum mit wöchentlichen Aufgabenstellungen und deren Besprechung zur aktiven und eigenständigen Auseinandersetzung mit den Lehrinhalten.

Prüfung

Prüfungsnummer

3970020

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Programmieren 2

Prüfungsform

Elektronische Prüfung, 90 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Die Themenbereiche aus der Vorlesung Programmieren 1

Inhalte des Moduls

Um keinen unnötigen Bruch gegenüber Programmieren 1 entstehen zu lassen, wird die Vertiefung der OOP an der Java-Plattform demonstriert.

Einige zentrale Themenbereiche:

- Containerklassen: Das Collection-API
- Java funktional: Lambdas und Stream-API
- Ein-, Ausgabe: Streams
- Graphische Benutzeroberflächen: JavaFX
- Nebenläufigkeit: Threads
- Netze und Verteilung: Sockets und RMI
- Softwarequalität und -struktur: Unit-Testing, Refactoring, Design Patterns, etc.

Im Praktikum werden die Vorlesungsinhalte in einem unter Anleitung schrittweise vorangetriebenen Projekt angewendet und vertieft.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- verfestigen und vertiefen das objektorientierte Programmierparadigma.
- erhalten Einblick in den Aufbau und die Verwendung von umfangreichen APIs für verschiedene Anwendungszwecke.
- werden für die Wichtigkeit der internen Qualität von Software sensibilisiert.
- erlangen Kern-Kompetenzen für die Mitwirkung an der Erstellung und Weiterentwicklung von produktiver Anwendungssoftware.

Literaturliste

Aus der Vielzahl von Java-Büchern als Begleitung zur Vorlesung besonders geeignet:

Goll. J. al: Java als erste Programmiersprache. Teubner.

Ullenboom Christian: Java ist auch eine Insel. Galileo Computing. Online Ausgabe:
<http://www.tutego.com/javabuch/online.htm>

2.2 Software-Engineering 2

Name / engl.

Software-Engineering 2 / Software-Engineering 2

Kürzel

SE2

Verantwortlicher

Prof. Dr. Phillip Heidegger

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Software-Engineering 2 (2 SWS),

Praktikum Software-Engineering 2 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970040

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Software-Engineering 2

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Software-Engineering 1 (empfohlen), Modul Programmieren 1 (empfohlen)

Inhalte des Moduls

Die Vorlesung Software Engineering II konzentriert sich auf die Vermittlung der in der Entwurfsphase der Softwareentwicklung notwendigen Techniken und Methoden. Hierzu werden zuerst Entwurfsprinzipien von Software vorgestellt.

Ein weiterer Aspekt der Veranstaltung ist die Modellierung von Software mithilfe von UML. Es werden die wichtigsten UML-Diagramme besprochen und deren Einsatz in der Softwareentwicklung erläutert. Anschließend werden dann unter Zuhilfenahme der UML häufig eingesetzte Entwurfsmuster vorgestellt und mit den zu Beginn der Veranstaltung besprochenen Entwurfsprinzipien in Verbindung gebracht.

Anhand eines Miniprojektes werden diese Techniken angewendet. Dabei werden auch Themen zur agilen Arbeitsorganisation und -planung eingeführt sowie Aspekte der Teamarbeit erläutert.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- verstehen die grundlegenden Softwareentwurfsprinzipien, können diese erkennen und anwenden
- können Sachverhalte mit UML-Diagrammen beschreiben
- kennen wichtige Entwurfsmuster und können diese anwenden
- können Arbeitsschritte aufteilen und im Team entwickeln
- können Softwareentwicklungstechniken in realistischen Szenarien einsetzen

Literaturliste

- Chris Rupp und Stefan Queins, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 2012, ISBN-13: 978-3446430570
- Florian Siebler, Design Patterns mit Java, Eine EINFÜHRUNG in ENTWURFSMUSTER, 2014, Print-ISBN: 978-3-446-43616-9, E-Book-ISBN: 978-3-446-44111-8
- Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, 2015, ISBN: 0-201-63361-2
- Ludwig und Lichter, Software Engineering: Grundlagen, Menschen, Prozesse, Techniken; dpunkt.verlag 3. korrigierte Auflage 2013, ISBN-13: 978-3864900921
- Boris Gloger, Scrum: Produkte zuverlässig und schnell entwickeln, Carl Hanser Verlag GmbH, 2008, ISBN-13: 978-3446414952
- Ester Derby, Diana Larson, Agile Retrospectives: Making Good Teams Great, O'Reilly 2004, ISBN-13: 978-0977616640

2.3 Mathematik 2

Name / engl.

Mathematik 2 / Mathematics 2

Kürzel

MAT2

Verantwortlicher

Prof. Dr. Stefan Glasauer

Lehrsprache

Deutsch

Fakultät

Fakultät für Geistes- und Naturwissenschaften

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,

Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Mathematik 2 (5 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

3970060

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 90 Minuten, Hilfsmittel: Taschenrechner

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Mathematik 1 (empfohlen)

nützlich für

Die Themenbereiche sind für die folgenden Module relevant:

- Programmieren 2-3
 - Grundlagen der Informatik 2
 - Datenkommunikation
 - Statistik
 - Datenbanken
 - Betriebswirtschaftslehre
 - Numerische Mathematik
-

Inhalte des Moduls

- Integralrechnung
- Zahlentheorie
- Lineare Gleichungssysteme
- Matrizen und Vektoren
- Komplexe Zahlen
- Differenzialrechnung in mehreren Variablen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- erlangen sowohl Rechenfertigkeiten als auch ein grundlegendes Verständnis von mathematischen Methoden und Zusammenhängen.
- sind in der Lage, mathematische Methoden in der Informatik und ihren Anwendungsgebieten zur Problemlösung einzusetzen.
- sind ferner in der Lage, sich im Studium und Beruf benötigte mathematische Hilfsmittel anhand von Lehrbuchliteratur nach Bedarf selbst anzueignen.

Literaturliste

Teschl/Teschl: Mathematik für Informatiker 1 und 2, Springer.

Hartmann: Mathematik für Informatiker, Vieweg + Teubner.

2.4 Grundlagen der Informatik 2

Name / engl.

Grundlagen der Informatik 2 / Fundamentals of Computer Sciences 2

Kürzel GDI2	Verantwortlicher Prof. Dr.-Ing. Honorary Doctor of ONPU Thorsten Schö- ler
Lehrsprache Deutsch	Fakultät Fakultät für Informatik
Verwendbarkeit Informatik Bachelor	Dauer / Angebot ein Semester, jeweils im Sommersemester
Arbeitsaufwand / Zusammensetzung SWS: 5, CPs: 6, Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h	
Lehrveranstaltungen Grundlagen der Informatik 2 (4 SWS) Praktikum Grundlagen der Informatik 2 (1 SWS)	
Lehr-/Lernmethoden Seminaristischer Unterricht, Praktikum	
Prüfung	
Prüfungsnummer 3970080	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
Zulassungsvoraussetzung zur Prüfung Praktikum Grundlagen der Informatik 2	
Prüfungsform Klausur, 60 Minuten, Hilfsmittel: nicht-prog. Taschenrechner(kein Smartphone) (10% Bonuspunkte durch Studienarbeit in Praktikum möglich)	
Zusätzliche Informationen	
hilfreiche Voraussetzungen Modul Grundlagen der Informatik 1 (empfohlen)	

Inhalte des Moduls

Einleitung

- Algorithmen und Datenstrukturen
- Komplexitätstheorie
- Primzahlen, Zufallszahlen

Lineare Datenstrukturen

- Lineare Liste
- Suche in Zeichenfolgen
- Stack-basierte Algorithmen

Bäume

- Baumstrukturen
- Binärbaum
- Heap
- Klassifikation von Sortierverfahren

B-Baum-Familie

- “Paging” von Binärbäumen
- Erweiterungen

Graphen

- Grundbegriffe
- Elementare Graphenalgorithmen
- Algorithmen auf gewichteten Graphen
- Fluss in Netzwerken

Gestreute Speicherung

- Hash-Algorithmus
- Kollisionsauflösung
- Erweiterbares Hashing

- Kryptographische hash-Funktionen

Externe Medien

- Dateikonzepte
- Nebenläufige Verarbeitung
- Indexsequentielle Speicherung
- Indizierte Dateien

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden beherrschen die Grundlagen von Algorithmen und Datenstrukturen.

Die Studierenden erlangen Kenntnisse grundlegender Datenstrukturen und Verarbeitungstechniken unter Einbeziehung externer Speichermedien und die Fähigkeit, sie anzuwenden (Komplexität und Effizienz von Algorithmen; Suchen und Sortieren; Lineare und Dynamische Strukturen; Bäume; Graphen; Algorithmen auf externen Medien; Anwendungen).

Im Praktikumsteil werden Übungsaufgaben zu den wesentlichen in der Vorlesung systematisch vorgestellten Algorithmen gemeinsam erarbeitet.

Literaturliste

G. Saake and K.-U. Sattler, Algorithmen und Datenstrukturen: eine Einführung mit Java. Heidelberg: dpunkt, 2014.

H. Herold, B. Lurz, and J. Wohlrab, Grundlagen der Informatik, Auflage: 2., aktualisierte Auflage. München: Pearson Studium, 2012.

T. Häberlein, Praktische Algorithmik mit Python. München: Oldenbourg, 2012.

J. V. Guttag, Introduction to Computation and Programming Using Python, Auflage: Exp Rev. Cambridge, Massachusetts: The MIT Press, 2013.

Weiterführende / ergänzende Literatur

U. Schöningh, Ideen der Informatik: Grundlegende Modelle und Konzepte der Theoretischen Informatik, Korrigierte Auflage. München: Oldenbourg Wissenschaftsverlag, 2008.

2.5 Datenkommunikation

Name / engl.

Datenkommunikation / Fundamentals of Data Communications

Kürzel

DAKO

Verantwortlicher

Prof. Dr. Rolf Winter

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Datenkommunikation (3 SWS)

Praktikum Datenkommunikation (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970100

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, Hilfsmittel: Taschenrechner

Inhalte des Moduls

Funktionsweise und Aufbau der Internet-Architektur und seiner Prinzipien und Protokolle insbesondere:

- Protokolle der Anwendungsschicht (wie HTTP und DNS)
- Transport-Protokolle (wie TCP und UDP)
- Routing-Protokolle (link state und distance vector)
- Protokolle der Sicherungsschicht (z.B. Ethernet)
- Arbeitsweise von Kernkomponenten des Internets (Switches, CDNs, NAT, uvm.)
- Aspekte der Netzsicherheit (z.B. Paketfilter)
- Schlüsselprinzipien des Internets (Zuverlässige Datenübertragung, Staukontrolle etc.)
- Umgang mit Standardwerkzeugen (Software) im Bereich Netzwerke
- Netzeinrichtung, Wartung und Fehlerdiagnose

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden kennen die Schlüsselprotokolle des Internets und können deren Aufgaben und Funktionsweise im Detail erklären. Sie wissen welche Funktionen der Internet-Architektur wie und wo im Netz implementiert sind. Auch die komplexen Zusammenhänge zwischen Protokollen und Mechanismen im Internet können Studierende beschreiben.

Darüber hinaus können die Studierenden ihr erlerntes Wissen auch praktisch bei der Entwicklung von vernetzten Anwendungen oder der Einrichtung und Wartung von Netzen einsetzen. Das Praktikum befähigt Studierende mit Standardwerkzeugen Anwendungen und Netze zu analysieren und einzurichten.

Literaturliste

Kurose, J.F./ Ross, K.W.: Computernetzwerke, 6. Auflage, Pearson Studium, 3/2014, ca. 900 Seiten, ISBN 978-3-8689-4237-8

3 Informatik Bachelor - 3. Semester

3.1 Programmieren 3

Name / engl.

Programmieren 3 / Programming 3

Kürzel

PGR3

Verantwortlicher

Prof. Dr. Peter Rösch

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 8,

Präsenzzeit: 90 h, Selbststudium: 150 h, Gesamtaufwand: 240 h

Lehrveranstaltungen

Programmieren 3 (4 SWS)

Praktikum Programmieren 3 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum, teilweise mit Teamarbeit.

Prüfung

Prüfungsnummer

3970110

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Programmieren 3

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

- Objektorientierte Softwareentwicklung
- Grundlagen der Vektorrechnung und Analysis

Inhalte des Moduls

Effiziente Software-Entwicklung mit Python

- Python – Einführung
- Interaktive Software-Entwicklung mit Jupyter Notebooks
- Automatisierung von Tests
- Systematische Optimierung
- Grafische Benutzerschnittstellen

Fortgeschrittene Programmiertechniken mit Python

- Nebenläufigkeit
- Entwurfsmuster
- Integration heterogener Komponenten
- Wissenschaftliche Anwendungen
- Verteilte Anwendungen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Die syntaktischen Konstrukte der Programmiersprache Python zu beschreiben.
- Vorgegebenen Quellcode bezüglich Effizienz und Qualität zu klassifizieren.
- Anwendungen bezüglich des Verbrauchs von Rechenzeit und Speicher zu optimieren.
- Die Implementierungen von Algorithmen mittlerer Komplexität in verschiedenen Programmiersprachen zu vergleichen.
- Aufgabenstellungen durch die geschickte Kombination existierender Komponenten zu lösen.
- Probleme mittlerer Komplexität in Teilprobleme zu zerlegen.
- Software-Komponenten für die Lösung von Problemen mittlerer Komplexität selbst zu entwickeln, zu testen und zu dokumentieren.

Literaturliste

Johannes Ernesti, Peter Kaiser: Python3 – Das umfassende Handbuch, 5. Auflage, Rheinwerk Computing (2017)

Bernd Klein: Einführung in Python 3, Hanser (2014)

Mark Pilgrim: Python 3 – Intensivkurs, Springer (2010)

Dusty Phillips: Python 3 Object-Oriented Programming, 3. Auflage, Packt Publishing (2018)

Eric Freeman, Elisabeth Freeman: Entwurfsmuster von Kopf bis Fuß, O'Reilly (2015)

Mark Summerfield: Rapid GUI Programming with Python and Qt - The definitive Guide to PyQt Programming, Prentice Hall (2015)

Python-Homepage: <https://www.python.org/>

3.2 Software-Engineering 3

Name / engl.

Software-Engineering 3 / Software-Engineering 3

Kürzel

SE3

Verantwortlicher

Prof. Dr. Alexandra Teynor

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Software-Engineering 3 (2 SWS)

Praktikum Software-Engineering 3 (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970120

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Software-Engineering 3

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Software-Engineering 2 (empfohlen), Modul Programmieren 2 (empfohlen)

Inhalte des Moduls

Den Schwerpunkt dieses Moduls bilden verschiedene Verfahren, Qualität von Software sicherzustellen. Hierfür wird zuerst der Begriff der Qualität eingehend erklärt.

Der Zusammenhang zwischen Architektur und Qualität von Software wird besprochen und im Praktikum durch Diskussionen vertieft. Zwei unterschiedliche Sichten auf Architektur, die Macro- und die Mikroarchitektur werden vorgestellt. Um beide Sichten besser greifbar zu machen werden aus beiden Perspektiven einige Themen ausführlicher behandelt.

Bei der Makroarchitektur werden u.a. Architekturpattern behandelt. Im Bereich Mikroarchitektur wird der Unterschied zwischen synchroner und asynchroner Programmierung und die Auswirkungen auf die Softwareentwicklung besprochen.

Außerdem wird in dem Modul behandelt, wie Dokumentation von Architektur den Softwareentwicklungsprozess unterstützen kann.

Es wird dargestellt wie mithilfe von unterschiedlichen Testverfahren die Qualität von Software verbessert werden kann. Gütekriterien für Softwaretests werden behandelt.

Als letztes Thema behandelt das Modul die Frage, wie während der Entwicklung aufgestaute technische Schulden beseitigt werden können (Refactoring).

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden können

- die Bedeutung des Begriffs Qualität von Software erläutern.
- die Bedeutung des Begriffs Architektur von Software erläutern.
- den Zusammenhang zwischen Qualität und Architektur darlegen.
- Mikro- und Makroarchitektur unterscheiden.
- Architekturpattern in Softwareprojekten einsetzen und entscheiden, welche Pattern in welchen Szenarien Sinn machen.
- asynchron arbeitenden Software entwickeln.
- Vor- und Nachteil von synchroner und asynchroner Software einschätzen.
- die Architektur von Software dokumentieren.
- mithilfe von Tests die Qualität von Software überprüfen und festgelegte Forderungen an die Qualität von Software sicherstellen.
- die Güte von Tests für Softwaresysteme einschätzen.
- Methoden anwenden, um Software strukturell zu verbessern.

Literaturliste

Fowler, 1999: Refactoring, Improving the Design of Existing Code, ISBN-13: 978-0-201-48567-7

Liggesmeyer, 2009: Software-Qualität: Testen, Analysieren und Verifizieren von Software, ISBN-13: 978-3-827-42056-5

Ludewig und Lichter, 2013: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, ISBN-13: 978-3-864-90092-1

Myers, Sandler und Badgett, 2011: The Art of Software Testing, ISBN-13: 978-1-118-03196-4

Vogel u.a., 2009: Software-Architektur: Grundlagen – Konzepte – Praxis, ISBN-13: 978-3-827-41933-0

Zeller, 2009: Why Programs Fail, ISBN-13: 978-3-898-64620-8

Zörner, 2015: Software Architekturen Dokumentieren und Kommunizieren, ISBN-13: 978-3-446-44348-8

3.3 Statistik

Name / engl.

Statistik / Statistics

Kürzel

STAT

Verantwortlicher

Dr. Torsten Straßer

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.
Das Modul wird regelmäßig im Wintersemester angeboten.

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,
Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Statistik (5 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen

Prüfung

Prüfungsnummer

3970130

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 90 Minuten, Hilfsmittel: Skript, eigene Notizen, Statistikbuch, Notebook, Tablet, Statistiksoftware (z. B. SAS JMP, R, o.ä.), Internetzugang

Inhalte des Moduls

- Einleitung
 - Grundbegriffe der Datenerhebung
 - Einführendes zu R und RStudio
- Deskriptive Statistik
 - Häufigkeiten
 - Lage und Streuung
 - Konzentration
 - Zwei Merkmale
 - Korrelation
 - Lineare Regression
- Wahrscheinlichkeitstheorie
 - Kombinatorik
 - Zufall und Wahrscheinlichkeit
 - Zufallsvariablen und Verteilungen
 - Verteilungsparameter
- Induktive Statistik
 - Stichproben
 - Schätz- und Testfunktionen
 - Punkt-Schätzung
 - Intervall-Schätzung
 - Signifikanztests

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden werden in die statistische Methodenlehre eingeführt und beherrschen die wichtigsten Methoden der Aufbereitung und Analyse und Auswertung von Daten.

Literaturliste

Bamberg, Günter; Baur, Franz; Krapp, Michael: Statistik, Oldenbourg Wissenschaftsverlag, 17.Aufl. 2012

Fahrmeir, Ludwig; Künstler, Rita; Pigeot, Iris; Tutz, Gerhard: Statistik, Springer, 7. Aufl. 2012

3.4 Datenbanken

Name / engl.

Datenbanken / Database Management Systems

Kürzel

DB

Verantwortlicher

Prof. Matthias Kolonko, Ph.D. (ONPU)

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 8,

Präsenzzeit: 90 h, Selbststudium: 150 h, Gesamtaufwand: 240 h

Lehrveranstaltungen

Datenbanken (4 SWS)

Datenbanken Praktikum (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Übungen, Praktikum

Prüfung

Prüfungsnummer

3970140

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Zulassungsvoraussetzung zur Prüfung

Praktikum Datenbanken

Prüfungsform

Klausur, 90 Minuten, keine Hilfsmittel

Alternativ zur Klausur können auch 3 Studienarbeiten abgegeben werden, die zu einer Gesamtnote zusammengefasst werden. Mit Abgabe der 3 Studienarbeiten ist ein Wechsel zur Prüfungsart Klausur im jeweiligen Semester nicht mehr möglich.

Zusätzliche Informationen

nützlich für

Modul ist auch von Wirtschaftsinformatik (Bachelor) belegbar.

Inhalte des Moduls

Die Vorlesung befasst sich mit drei zentralen Punkten im Umfeld der Datenbanksysteme. Den erste Kernpunkt stellt die semantische Datenmodellierung und der systemunabhängige Datenbankentwurf dar. Danach folgt die Umsetzung anhand relationaler Datenbanksysteme mittels SQL. Im Anschluss wird vertiefend auf die Normalformentheorie eingegangen. Hierbei werden sowohl praktische als auch theoretische Aspekte beleuchtet. Die Architektur eines Datenbank-Management-Systems und geeignete physische Datenstrukturen werden an einem verbreiteten RDBMS aufgezeigt.

Es wird eine Instanz einer relationalen Datenbank für SQL-Übungen zur Verfügung gestellt. Im Praktikum werden anhand eines selbst gewählten Themas Entwurf und Realisierung einer passenden Datenbankstruktur geübt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Absolvierung des Moduls können die Studierenden

- die grundlegende Architektur und Funktionsweise eines Datenbanksystems beschreiben.
- Analyse und Datenmodellierung (konzeptueller und logischer Datenbankentwurf) durchführen.
- die grundlegenden Operationen von SQL nennen.
- Datenstrukturen und Anfragen an eine relationale Datenbanke mittels SQL (DDL & DML) realisieren.
- die Analyse und Normalisierung eines logischen Datenmodells anhand der Normalformentheorie anwenden.

Gewichtung der Einzelleistung in der Gesamtnote

Klausur (100%) oder 3 Studienarbeiten (jeweils mit gleicher Gewichtung)

Literaturliste

- R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems* (Pearson 2020, ISBN: 1-292-09761-2)
- S. Müllenbach, L. Kern-Bausch, M. Kolonko: Conceptual Modeling Language AGILA MOD
in Herald of Advanced Information Technology, vol. 2, no. 4, pp. 246-258, Dez. 2019
(ISSN: 2663-0176 – DOI: 10.15276/hait.04.2019.1)
- M. Kolonko, S. Müllenbach, E. Arsirii, B. Trofymov: *Extensions to the Conceptual Modeling Language AGILA MOD*
in Proceedings of the VI. Ukrainian-German conference „Informatics. Culture. Technology“, Odessa, Sept. 2018, pp. 38-39
- L. Kern-Bausch, M. Jeckle: Informationsmodellierung und logischer Datenbankentwurf, Kapitel 14.2
in Taschenbuch der Informatik (U. Schneider und D. Werner), 4. Auflage, Fachbuchverlag Leipzig im Carl Hanser Verlag, 2001,
ISBN: 3-446-21753-3
- P. Sauer: Informationsmodellierung, Kapitel 2
in Taschenbuch Datenbanken (T. Kudraß), 2. Auflage, Fachbuchverlag Leipzig im Carl Hanser Verlag, 2015,
ISBN: 978-3-446-43508-7
- Vorlesungsunterlagen von Prof. Dr. Sabine Müllenbach unter <https://ohs.informatik.hsaugsburg.de:4443/web/bine>
(Anmeldung mit RZ-Login)

4 Informatik Bachelor - 4. Semester

4.1 Numerische Mathematik

Name / engl.

Numerische Mathematik / Numerical Mathematics

Kürzel

NMAT

Verantwortlicher

Prof. Dr. Stefan Glasauer

Lehrsprache

Deutsch

Fakultät

Fakultät für Geistes- und Naturwissenschaften

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Numerische Mathematik (3 SWS)

Praktikum Numerische Mathematik (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970150

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 90 Minuten, Hilfsmittel: Taschenrechner,

4 DIN-A4-Seiten handgeschrieben

Zusätzliche Informationen

hilfreiche Voraussetzungen

Analysis:

- Ableitung, Taylorentwicklung, Integration elementarer Funktionen

Lineare Gleichungssysteme:

- Gauss-Verfahren, Lösbarkeit, Struktur der Lösungsmengen

Matrizen, Determinanten:

- Rechnen mit Matrizen, Verwendung der inversen Matrix zur Lösung eines Gleichungssystems, Bedeutung der Determinante, Eigenschaften einer Determinante, Lösung eines Gleichungssystems mit Hilfe der Cramerschen Regel.

Vektoren:

- Vertrautheit mit reellen Vektoren, insbesondere auch Skalarprodukt und Norm
-

Inhalte des Moduls

- Gleitpunktarithmetik
- Numerische Nullstellenbestimmung
- Lineare Gleichungssysteme
- Numerische Lösung nichtlinearer Gleichungssysteme
- Interpolation
- Kleinste-Quadrate-Approximation
- Diskrete Fourier-Transformation
- Numerische Integration
- Numerik gewöhnlicher Differenzialgleichungen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden...

- KENNEN: wichtige Phänomene des numerischen Rechnens, Themengebiete der numerischen Mathematik, Anwendungsbeispiele.
- VERSTEHEN: zentrale Lösungsideen und Algorithmen aus ausgewählten Themenbereichen der numerischen Mathematik.
- KÖNNEN: Ausführen von Algorithmen an einfachen Problemen, Diskussion der Methodenwahl und der Ergebnisse.

Literaturliste

Knorrenschild: *Numerische Mathematik*, Hanser 2017.

Schwarz, Köckler: *Numerische Mathematik*, Teubner 2011.

4.2 Rechnerstrukturen 2

Name / engl.

Rechnerstrukturen 2 / Computer Structures 2

Kürzel

REC2

Verantwortlicher

Prof. Dr. Michael Strohmeier

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Rechnerstrukturen 2 (4 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht und begleitende Übungen zur Anwendung und Vertiefung der erlernten analytischen und quantitativen Verfahren der Rechnerarchitektur.

Prüfung

Prüfungsnummer

3970160

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, Hilfsmittel: Taschenrechner, 2 DIN-A4-Seiten handgeschriebene Formelsammlung

Zusätzliche Informationen

hilfreiche Voraussetzungen

Modul Rechnerstrukturen 1

Inhalte des Moduls

Das Modul Rechnerstrukturen 2 erweitert und vertieft die Kenntnisse von Rechnerarchitekturen und deren Organisationsformen aus vorangegangenen Modulen wie Rechnerstrukturen 1 und wählt dazu eine klassifizierende, quantitative und analytische Vorgehensweise. Im Einzelnen werden folgende Themenkreise behandelt:

- Mooresches Gesetz und technologische Grenzen
- Rechnerklassifikation und –evolution
- Relevante Prozessorarchitekturen (Universalrechner, Pipeline-Prozessor, Superskalarprozessor, Multithreading, Multicore-Architekturen, Alternative Rechnerarchitekturen)
- Rechenwerke und Leitwerke
- Rechner-Leistungsbewertung
- Rechnerentwurf und Mikroelektronik
- Energieeffizienz in IT-Systemen
- Befehlssatzarchitekturen (ISA)
- Mikroarchitekturen
- Cache und Hauptspeicher
- Bussystem, Interconnect-Strukturen und Chipsätze
- Parallelrechner

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul Rechnerstrukturen 2 sind die Studierenden in der Lage Rechner zu klassifizieren und Struktur-, Organisations- und Implementierungsprinzipien aller gängigen Rechnerklassen und Prozessorarchitekturen zu verstehen und zu vergleichen. Sie können vorgegebene Rechnerarchitekturen auf der Mikroarchitektur-, Befehlssatz- und Systemebene analysieren.

Sie wissen, wie Prozessoren und Prozessorkerne mit dem Speicher / Bus-System / Interconnect-System zusammenwirken und sind in der Lage, grundlegende Leistungsbewertungen von Rechnersystemen vorzunehmen.

Die Studierenden können sich kritisch mit der Thematik des Rechnerentwurfs und den für die Prozessorentwicklung erforderlichen Mikroelektronik-Grundlagen und der technologischen Evolution auseinandersetzen.

Sie entwickeln auch Grundkenntnisse, um Fragen des energieeffizienten Entwurfs und Betriebs von Rechnersystemen kompetent zu beantworten.

Literaturliste

Hennessy J.L., Patterson D.A. A New Golden Age for Computer Architecture. In: Communications of the ACM 62, 2 Jg. (2019), S. 48-60

Hennessy J.L., Patterson D.A. Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann, 2017

Märting C.: Multicore Processors: Challenges, Opportunities, Emerging Trends. Embedded World Conference 2014, Weka Fachmedien, 2014

Patterson D.A., Hennessy J.L. Computer Organization and Design MIPS Edition: The Hardware/Software Interface, Revised 6th Edition, Morgan Kaufmann, 2020

Patterson D.A., Hennessy J.L. Computer Organization and Design RISC-V Edition: The Hardware/Software Interface, Revised 2nd Edition, Morgan Kaufmann, 2021

4.3 Betriebssysteme

Name / engl.

Betriebssysteme / Operating Systems

Kürzel

BSYS

Verantwortlicher

Prof. Dr. Volodymyr Brovko

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,

Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Betriebssysteme (3 SWS)

Praktikum Betriebssysteme (1 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Praktikum um den Stoff einzuüben.

Prüfung

Prüfungsnummer

3970170

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Inhalte des Moduls

- Grundstrukturen und Arbeitsweisen von Betriebssystemen
- Prozesse, Threads und Scheduling
- Synchronisation und Kommunikation
- Speicherverwaltung
- Ein-/Ausgabe
- Dateisysteme
- Sicherheit in Betriebssystemen

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden sind in der Lage,

- die Grundstrukturen von Betriebssystemen zu nennen und zu identifizieren.
- die wesentliche Arbeitsweise der Prozessverwaltung, der Speicherverwaltung sowie des Ein-/Ausgabesystems eines Betriebssystems zu erklären.
- den Aufbau und die Arbeitsweise eines Gerätetreibers zu erklären.
- POSIX-Systemfunktionen zu benutzen, um systemnahe Software zu implementieren.
- geläufige Synchronisationsmechanismen richtig anzuwenden, um parallele Anwendungen korrekt zu implementieren.
- die Effizienz von Software im Hinblick auf die Nutzung von Betriebssystem-Ressourcen zu analysieren und zu beurteilen.

Literaturliste

William Stallings: Operating Systems - Internals and Design Principles, 9. Auflage, Pearson, 2018, ISBN: 9780134700069

Andrew S. Tanenbaum, Herbert Bos: Modern Operating Systems, 4. Auflage, Pearson, 2015, ISBN: 978-1-292-06142-9, 1-292-06142-1

Jürgen Quade, Eva-Katharina Kunst: Linux-Treiber entwickeln, 4. Auflage, dpunkt.verlag, 2016

4.4 Projektarbeit 1

Name / engl.

Projektarbeit 1 / Project Work 1

Kürzel

PA1

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 6, CPs: 8,

Präsenzzeit: 90 h, Selbststudium: 150 h, Gesamtaufwand: 240 h

Lehrveranstaltungen

Projektarbeit 1 (6 SWS)

Lehr-/Lernmethoden

Studierende erarbeiten in Kleingruppen weitestgehend eigenständig Lösungen zu einem praxisorientierten Thema aus dem IT-Bereich.

Ziel ist es einen Projektablauf möglichst realitätsnah mit allen Facetten abzubilden. Die Projektthemen werden von Prüfungsberechtigten der Fakultät für Informatik vergeben und umfassen einen praktischen Teil (Software/Hardware), eine Dokumentation (Studienarbeit) und eine Präsentation.

Die Präsentation findet in der Regel im Rahmen eines Projekttagess statt. Die Abstimmung mit dem Projektsteller erfolgt in regelmäßigen persönlichen Treffen und über elektronische Kanäle. Die Bearbeitung ist nicht notwendigerweise an die Vorlesungszeit gebunden.

Prüfung

Prüfungsnummer

3970180

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- Projektarbeit, 10-30 Seiten, 80%
- Präsentation, 20-40 Minuten, 20%

Zusätzliche Informationen

hilfreiche Voraussetzungen

Grundlagen des Softwareengineering und objektorientierte Programmiertechniken wie sie in den Modulen SWE1, PRG1 und PRG2 vermittelt werden.

Inhalte des Moduls

Die Studierenden führen in kleinen Teams IT-Projekte durch. Zu den Aufgaben der Studierenden zählen Projektmanagement, sowie je nach Projekt die klassische oder agile Softwareentwicklung, Dokumentation und Präsentation der Ergebnisse.

Zum Beginn des Moduls findet ein Seminar zur Vermittlung der Grundlagen der Teamarbeit und dem Umgang mit Teamkonflikten statt. Alle Mitglieder der Gruppen sollen zu Beginn der Arbeit ein Minimum an Kenntnissen von Teamprozessen haben, um den eigenen Lernerfolg und den Lernerfolg der Teammitglieder sicherzustellen. Aus diesem Grund ist die Teilnahme an dem Seminar verpflichtend.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Teamprozesse zu verstehen und typische Teamkonflikte zu lösen.
- Software- und Hardware-Entwicklungsprojekte im Team hinsichtlich Zeit, Aufwände und Ressourcen zu planen und durchzuführen.
- Agile oder klassische Projektmanagementmethoden praktisch anzuwenden.
- Auswahl geeigneter Methoden und selbstständiges Erlernen neuer Techniken.
- Projektergebnisse verständlich zu dokumentieren und ansprechend zu präsentieren.

Literaturliste

Projektspezifische Literatur wird vom Betreuer vor Beginn des Projektes bekanntgegeben.

5 Informatik Bachelor - 5. Semester

5.1 Praktische Tätigkeit (Praxissemester)

Name / engl.

Praktische Tätigkeit (Praxissemester) / **Practical Term**

Kürzel

PRAX

Verantwortlicher

Praktikantenbeauftragte

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

ECTS: 20, Gesamtaufwand: 20 Wochen

Lehrveranstaltungen

Praktische Tätigkeit (20 Wochen)

Lehr-/Lernmethoden

Praktische Tätigkeit

Prüfung

Prüfungsnummer

3970240

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Praxisbericht, 20-50 Seiten

Zusätzliche Informationen

hilfreiche Voraussetzungen

Die Aufnahme der praktischen Tätigkeit und die Teilnahme am Praxisseminar ist zulässig, wenn mindestens 80 ECTS nachgewiesen wurden.

Inhalte des Moduls

Besonderheit:

Als Besonderheit des Studiums an bayerischen Hochschulen bieten wir Ihnen ein in das Studium integriertes, gesetzlich vorgeschriebenes praktisches Studiensemester, in welchem der Schwerpunkt der Wissensvermittlung in die Praxis hinaus verlegt wird. Während des Praxissemesters behalten Sie Ihren Status als Studentin oder Student bei, die praktische Ausbildung wird durch begleitende Unterrichtsveranstaltungen an der Hochschule ergänzt und vertieft.

Zuständig für die formale Abwicklung des Praktikums ist das Praktikantenamt. Lesen Sie deshalb bitte auch den Leitfaden für die praktischen Studiensemester des Praktikantenamtes.

Neben dem Praktikantenamt steht Ihnen ein fachlicher Betreuer zur Seite. Sprechen Sie ihn bitte insbesondere dann möglichst frühzeitig an, wenn es mit Ihrer Praktischen Tätigkeit irgendwelche Probleme gibt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

- Anleitung zum selbstständigen und eigenverantwortlichen Arbeiten
- Einführung in das Berufsfeld durch möglichst selbstständige und eigenverantwortliche Mitarbeit
- Erweiterung und Vertiefung der Kenntnisse über organisatorische Problemlösungen im Betrieb
- Kenntnisse über Fragen der Berufsausübung wie Tätigkeitsmöglichkeiten, arbeitsrechtliche Formen und Arbeitsplätze
- Einblick in relevante Steuerverordnungen und soziale Absicherungen.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Falls notwendig wird die Literatur im Praktikumsbetrieb bekannt gegeben.

5.2 Praxis-Seminar

Name / engl.

Praxis-Seminar / Practical Term Mentoring

Kürzel PSEM	Verantwortlicher Professorinnen und Professoren der Fakultät für Informatik
Lehrsprache Deutsch	Fakultät Fakultät für Informatik
Verwendbarkeit Informatik Bachelor	Dauer / Angebot ein Semester, jeweils im Wintersemester
Arbeitsaufwand / Zusammensetzung SWS: 2, CPs: 2, Präsenzzeit: 30 h, Selbststudium: 30 h, Gesamtaufwand: 60 h	
Lehrveranstaltungen Praxisseminar (2 SWS)	
Lehr-/Lernmethoden Seminar	

Prüfung

Prüfungsnummer 3970250	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
Prüfungsform Präsentation, 15-30 Minuten	

Zusätzliche Informationen

hilfreiche Voraussetzungen

Die Aufnahme der praktischen Tätigkeit und die Teilnahme am Praxisseminar ist zulässig, wenn mindestens 80 ECTS nachgewiesen wurden.

Inhalte des Moduls

Studierende können

- die eigene Arbeit nach wissenschaftlichen Standards korrekt und nachvollziehbar präsentieren sowie Fragen beantworten.
- Präsentationen zu anderen Arbeiten verstehen und sich an fachlichen Diskussionen beteiligen.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden erweitern die Kompetenz für Präsentationen.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Literatur wird im Seminar bekannt gegeben.

5.3 Systemnahe Programmierung

Name / engl.

Systemnahe Programmierung / Systems programming

Kürzel

SNP

Verantwortlicher

Prof. Dr. Hubert Högl

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Wintersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,

Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Systemnahe Programmierung (5 SWS)

Lehr-/Lernmethoden

Vorlesung, Seminaristischer Unterricht, Praktikum

Prüfung

Prüfungsnummer

3970190

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Klausur, 60 Minuten, keine Hilfsmittel

Zusätzliche Informationen

nützlich für

Informatik Bachelor

Inhalte des Moduls

Einführung in die systemnahe Programmierung am Beispiel einer zeitgemässen Programmiersprache zur sicheren Programmierung (Rust).

- Klärung des Begriffs Systemprogrammierung
- Einsatzbereiche für Systemprogrammierung
- Erstellen von ausführbaren Programmen und Bibliotheken
- Grundlegende Sprachelemente (Variablen, Datentypen, Strings, Kontrollstrukturen)
- Eigentümerschaft (ownership), Referenzen und Borgen (borrowing)
- Kollektionen
- Fehlerbehandlung
- Generische Programmierung
- Tests
- Zeiger
- Nebenläufigkeit

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden...

- wissen, welche Bereiche die Systemprogrammierung umfasst.
- kennen die Grundlagen der Systemprogrammierung in der Programmiersprache Rust und können diese anwenden um eigene Programme zu schreiben.
- verstehen den Ablauf von Programmen auf der Maschinenebene und können dadurch die Sprachelemente von Systemprogrammiersprachen optimal anwenden.
- können die Verwendung der verschiedenen Speicherbereiche in einem Programm erklären (u.a. Stack und Heap) und verstehen dadurch die Techniken der Speicherverwaltung in Systemprogrammiersprachen.
- schreiben Testfälle zur Entwicklung von sicherer Software.
- benutzen sichere parallele Sprachkonstrukte zur Beschleunigung der Programmausführung auf Mehrkernprozessoren.

Die Inhalte der Vorlesung werden in einem begleitenden Praktikum an Hand von Übungsaufgaben vertieft.

Literaturliste

Steve Klabnick, Carol Nichols: The Rust Book, 2018 (freier Inhalt)

<https://doc.rust-lang.org/book>

Carlo Milanesi: Beginning Rust, Apress 2018.

Jim Blandy, Jason Orendorff, Leonora F.S. Tindall: Programming Rust, O'Reilly 2021.

6 Informatik Bachelor - 6. Semester

6.1 Praktikum DVA

Name / engl.

Praktikum DVA / Technical Applications of Data Processing

Kürzel PRAK	Verantwortlicher Prof. Dr. Hubert Högl Prof. Dr. Volodymyr Brovkov
Lehrsprache Deutsch	Fakultät Fakultät für Informatik
Verwendbarkeit Informatik Bachelor	Dauer / Angebot ein Semester, jeweils im Sommersemester
Arbeitsaufwand / Zusammensetzung SWS: 4, CPs: 5, Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h	
Lehrveranstaltungen Praktikum DVA (4 SWS)	
Lehr-/Lernmethoden Praktikum	

Prüfung

Prüfungsnummer 3970210	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
----------------------------------	--

Prüfungsform

Portfolioprüfung:

- Studienarbeit, 15-25 Seiten
- Präsentation, 20 Minuten

Zusätzliche Informationen

nützlich für

Das Software-Praktikum "DV-Anwendungen in der Technik" ist ein Pflichtfach im Studiengang Informatik, das auch in anderen Studiengängen angeboten werden kann.

Inhalte des Moduls

Dieses Praktikum dient zur Einführung und Vertiefung der Kenntnisse von technischen DV-Anwendungen, insbesondere auch der hardware- und systemnahen Programmierung. Es enthält Versuche zu diversen Hardwarekomponenten unter den Betriebssystemen Windows, LINUX und Echtzeitsystemen. Die Versuche befassen sich zum Teil mit Systemaufrufen und zum Teil mit der Programmierung von Bausteinen (parallele, serielle Schnittstelle, USB, Sprachausgabe, DSP, SCSI). Es müssen auch Interrupt Service Routinen und Treiber erstellt werden, z.B. für Windows oder Chipkartenleser. Im Unix-Bereich geht es um verteiltes Arbeiten im Netzwerk.

Hardwarenahe Programmierung erfordert zunächst eine Auseinandersetzung mit den technischen Beschreibungen der Hardware, auf die die Programme zugreifen sollen. Hinzu kommen Problemstellungen beim Zugriff auf die Hardware, die sich manchmal nur durch Versuche in den Griff bekommen lassen. In den Laboren werden Systeme bereitgestellt, die sich von der bekannten Hardware "zu Hause" unterscheiden und die hier "gefahrlos" untersucht werden können.

Rahmenbedingungen der Veranstaltung:

Es besteht freie Auswahl aus einem breiten Aufgaben-Angebot aus dem EDV-Alltag. Oftmals sind die Aufgaben recht allgemein gestellt und der Anspruch entsteht dadurch, dass ein Problem nicht nur experimentell, sondern auch in der Tiefe bearbeitet werden muss (z.B. Festplatte einbauen, Datentransfer auf Fremdrechner, Sound aktivieren).

Es gibt Versuche zu Themen wie Biometrie, Java Card, Sicherheit, Webservices, sowie zur Entwicklung von Programmen für technische Problemstellungen. Es werden nicht alle Versuche in jedem Semester angeboten. Fortlaufend werden neue Versuche vorbereitet. Die Studierenden können sich die Versuche auswählen.

Es ist eine Schwerpunktbildung möglich, indem eine Serie von aufeinander aufbauenden Versuchen oder eine Vertiefung durch Folgeversuche gewählt wird. Neue Versuche sind nach Absprache möglich.

Die Versuche werden im Labor für hardwarenahe Programmierung und im Labor für Prozeßrechentechnik durchgeführt.

Neben den (einseitigen) Versuchsanleitungen liegen für viele Versuche weitere, ausführliche Unterlagenmappen im Labor aus. In den Glasschränken des Labors finden Sie auch Kopien von Zeitschrift-Artikeln, Handbücher und Original-Literatur. Langfristig ist vorgesehen, Versuchsunterlagen elektronisch im Internet anzubieten.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden

- sind in der Lage Grundlagenwissen bei der Handhabung von exotischen oder aktuellen technischen Systemen einzusetzen.
- sind in der Lage, sich selbständig in neue Problemkreise einzuarbeiten: Problemorientiertes Lernen anstelle von Frontalunterricht.
- erlangen mehrere Wege zur Lösung der Probleme, es müssen Sackgassen erkannt werden und der Arbeitseinsatz sinnvoll gesteuert und in der Gruppe verteilt werden.

Bewertet wird eher die Vorgehensweise als das Ergebnis.

Gewichtung der Einzelleistung in der Gesamtnote

mit Erfolg abgelegt / ohne Erfolg abgelegt

Literaturliste

Wird in der Veranstaltung bekannt gegeben.

6.2 Projektarbeit 2

Name / engl.

Projektarbeit 2 / Project Work 2

Kürzel

PA2

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 8, CPs: 10,

Präsenzzeit: 120 h, Selbststudium: 180 h, Gesamtaufwand: 300 h

Lehrveranstaltungen

Projektarbeit 2 (8 SWS)

Lehr-/Lernmethoden

Studierende erarbeiten in Kleingruppen weitestgehend eigenständig Lösungen zu einem praxisorientierten Thema aus dem IT-Bereich.

Ziel ist es einen Projektablauf möglichst realitätsnah mit allen Facetten abzubilden. Die Projektthemen werden von Prüfungsberechtigten der Fakultät für Informatik vergeben und umfassen einen praktischen Teil (Software/Hardware), eine Dokumentation (Studienarbeit) und eine Präsentation.

Die Präsentation findet in der Regel im Rahmen eines Projekttagess statt. Die Abstimmung mit dem Projektsteller erfolgt in regelmäßigen persönlichen Treffen und über elektronische Kanäle. Die Bearbeitung ist nicht notwendigerweise an die Vorlesungszeit gebunden.

Prüfung

Prüfungsnummer

3970200

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- Projektarbeit, 10-30 Seiten, 80%
- Präsentation, 20-40 Minuten, 20%

Zusätzliche Informationen

hilfreiche Voraussetzungen

Solide Kenntnisse aus den wichtigsten Themenbereichen der Informatik, wie z.B. Algorithmen und Datenstrukturen, Programmieren, Datenbanken, Datenkommunikation, Rechnerstrukturen und Betriebssysteme. Die erworbenen Kenntnisse sollten bereits in der ersten Projektarbeit (Modul PA1) praktisch angewendet worden sein.

Inhalte des Moduls

Die Studierenden führen in kleinen Teams IT-Projekte durch. Zu den Aufgaben der Studierenden zählen Projektmanagement, sowie je nach Projekt die klassische oder agile Softwareentwicklung, Dokumentation und Präsentation der Ergebnisse.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Teamprozesse zu verstehen und typische Teamkonflikte zu lösen.
- Software- und Hardware-Entwicklungsprojekte im Team hinsichtlich Zeit, Aufwände und Ressourcen zu planen und durchzuführen.
- Agile oder klassische Projektmanagementmethoden praktisch anzuwenden.
- Auswahl geeigneter Methoden und selbstständiges Erlernen neuer Techniken.
- Projektergebnisse verständlich zu dokumentieren und ansprechend zu präsentieren.

Literaturliste

Projektspezifische Literatur wird vom Betreuer vor Beginn des Projektes bekanntgegeben.

6.3 Englisch

Name / engl.

Englisch / English Language Skills

Kürzel

ENG

Verantwortlicher

Prof. Dr. Svea Schauffler

Lehrsprache

Das Modul wird in englischer Sprache unterrichtet.

Fakultät

Fakultät für angewandte Geistes- und Naturwissenschaften

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.
Das Modul wird regelmäßig sowohl im Wintersemester als auch im Sommersemester angeboten.

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,
Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Englisch (4 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht in Gruppen von 20-25 Teilnehmern, interaktive, handlungsorientierte Sprachdidaktik, anwendungsorientierte Sprachlehre

Prüfung

Prüfungsnummer

3970220

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- Präsentation, 10-20 Minuten, 20%
- Mündliche Prüfung, 10-20 Minuten, 20%
- Klausur, 90 Minuten, keine Hilfsmittel, 60%

Zusätzliche Informationen

hilfreiche Voraussetzungen

Studierenden, die nicht über das Startniveau B1+ verfügen, wird dringend empfohlen vor der Veranstaltung Englisch 1/2 zunächst den Förderkurs Englisch oder einen anderen Englischkurs mit Grundlagenvermittlung zu besuchen.

Inhalte des Moduls

Die Lehrveranstaltung ist eine Kombination aus sprachlichem und digitalem Input durch den Lehrenden, eigenständigem Selbststudium und kommunikativem und anwendungsorientiertem Sprachunterricht, in den sich alle Teilnehmer einbringen. Die Veranstaltung findet in Gruppen von 20-25 Teilnehmern statt.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Der Pflichtkurs Englisch hat zum Ziel den sicheren Umgang mit der (Fach-)Sprache im Studium und im beruflichen Umfeld auf dem Niveau B2.

Dies geschieht durch handlungsorientierten und interaktiven Unterricht in der Fremdsprache. Die Schwerpunkte liegen auf wichtigen und nützlichen Fertigkeiten wie Textverständnis, Fachvokabular, schriftliche Korrespondenz, selbstsichere mündliche Kommunikation, Präsentieren und Verhandeln in der Fremdsprache.

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- Erfolgreich in Studium und Beruf schriftlich und mündlich auf Englisch zu kommunizieren
- Für ein englischsprachiges Fachpublikum Vorträge und Präsentationen zu halten
- In englischsprachigen Meetings und Verhandlungen zu argumentieren und sich interkulturell angemessen zu verhalten
- Sich für Jobs und Praktika im englischsprachigen Ausland zu bewerben
- Englische Texte zu verstehen und relevante Inhalte zusammenzufassen

Literaturliste

Die Literatur wird im Seminar bekannt gegeben.

6.4 Betriebswirtschaftslehre und DV-Recht

Name / engl.

Betriebswirtschaftslehre und DV-Recht / Basics of Business Administration and Computer Legislation

Kürzel

BWL+DVRE

Verantwortlicher

Prof. Dr. Norbert Gerth / Alma Lena Fritz, LL.M., LL.M.

Lehrsprache

Deutsch

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

ein Semester, jeweils im Sommersemester

Arbeitsaufwand / Zusammensetzung

SWS: 5, CPs: 6,

Präsenzzeit: 75 h, Selbststudium: 105 h, Gesamtaufwand: 180 h

Lehrveranstaltungen

Betriebswirtschaftslehre (3 SWS)

DV-Recht (2 SWS)

Lehr-/Lernmethoden

Seminaristischer Unterricht, Gastvorträge, Best Practices, Team-/Gruppenarbeit, Workshop, Fallbeispiele

Prüfung

Prüfungsnummer

3970230

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung:

- BWL: Klausur, 60 Minuten, keine Hilfsmittel, 60%
- DV-Recht: Klausur, 60 Minuten, Hilfsmittel: CompR (dtv Beck-Texte), 40%

Inhalte des Moduls

- Betriebswirtschaftslehre
 - Betriebswirtschaft heute und morgen - Unternehmen im Digitalen Wandel
 - Was Unternehmen von Startups lernen können?
 - Digitalisierung und Innovation
 - Entre- und Intrapreneure als neue Rollenvorbilder für IT-Profis
 - Digitale Schlüsseltechnologien und ihre Business-Potenziale
 - Von der Technik zum Digitalen Geschäftsmodell
 - Ansätze zur Beurteilung von Geschäftsideen
 - Ein marktorientiertes Innovationsmanagement als Schlüssel zum Erfolg
 - Kundenbedürfnisse und Kundennutzen
 - Das Konzept der Unique Selling Proposition
 - Business Model Generation: zentrale Ansätze zur Entwicklung innovativer Geschäftsmodelle
 - * Business Canvas / Lean Startup
 - * Design Thinking / Customer Development
 - * Agiles Projekt Management
 - Gründung und Führung eines Startups als Studierender bzw. Wissenschaftler
 - Finanzierung und Risikokapital für Startups
 - Programme und Förderungen für 'Digital Innovators'
- DV-Recht
 - Privatrecht
 - * Rechtsgeschäfte
 - * Allgemeines und Besonderes Schuldrecht
 - * Sachenrecht
 - Internetrecht
 - * Schutz von Domains
 - * Electronic Commerce
 - * Schadensersatzhaftung und Haftungsbeschränkung
 - Urheberrecht/Wettbewerbsrecht
 - * Grundbegriffe
 - * Schutz und Haftung
 - * Schadensersatzansprüche

- Datenschutz
 - * Merkmale und Grundbegriffe
 - * Anwendbare Rechtsvorschriften
 - * Telekommunikationsdatenschutz

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Studierende des Kurses sollten durch ihre Teilnahme ...

- in Betriebswirtschaftslehre
 - Verständnis entwickeln für die Bedeutung betriebswirtschaftlicher Fragestellungen in Hightech-Unternehmen
 - die Relevanz Digitaler Innovationen für die Zukunftsfähigkeit von Unternehmen einschätzen können
 - lernen, Technologiekonzepte aus Business-Sicht zu bewerten
 - relevante Kundenbedürfnisse verstehen
 - Möglichkeiten kennen lernen, um Technologiekonzepte zielgerichtet in praktikable Geschäftsmodelle zu übersetzen
 - einen Überblick erhalten hinsichtlich der zentralen betriebswirtschaftlichen Herausforderungen bei der Vermarktung von Innovationen:
 - * Marktsegmentierung und Zielgruppenabgrenzung
 - * Ableitung einer Value Proposition
 - * Entwicklung effektiver Vermarktungskonzepte (Distribution Channels und Customer Interaction)
 - * Kosten- und Umsatzplanung bzw. Finance
 - Einblicke erhalten in die grundlegenden Aufgaben bei der Gründung eines Startups (Businessplanung, Finanzierung, Rechtsform, Anmeldung etc.)
 - Möglichkeiten der Finanzierung von Hightech-Startups und Förderprogramme für Startups in BAY sowie das Gründernetzwerk am Campus der HSA kennen lernen
- in DV-Recht
 - die Grundzüge des Privatrechts und Grundzüge des DV-Rechts mit der Bedeutung des Datenschutzes sowie die praktische Bedeutung beherrschen.
 - Grundkenntnisse juristischer Fallbearbeitung im Vertragsrecht erlangen.
 - in der Lage sein, die erworbenen Kenntnisse im Beruf und Alltag anzuwenden.

Literaturliste

DIG

SCHALLMO et al. (Hrsg.) (2017): Digitale Transformation von Geschäftsmodellen: Grundlagen, Instrumente und Best Practices, Berlin/Wiesbaden: SpringerGabler

BWL / UF

MÜLLER (2013): Betriebswirtschaftslehre für Ingenieure, 2. Aufl., Berlin: Springer

INNO

GERTH 2015: IT-Marketing: Produkte anders denken - denn nichts ist, wie es scheint, 2. Aufl., Berlin u.a.: Springer

KASCHNY/NOLDEN/SCHREUDER (2015): Innovationsmanagement im Mittelstand - Strategien, Implementierung, Praxisbeispiele, Wiesbaden: GablerSpringer Fachmedien Wiesbaden

GRÜN

BayStartUP GmbH (Hrsg.) (2016): Handbuch zur Businessplan-Erstellung, 8. Aufl., Nürnberg

HOROWITZ (2014): The Hard Thing about Hard Things - Building a Business When There Are No Easy Answers, HarperBusiness

OSTERWALDER/PIGNEUR (2011): Business Model Generation - Ein Handbuch für Visionäre, Spielveränderer und Herausforderer, Campus Verlag

OSTERWALDER et al. (2015): Value Proposition Design - Entwickeln Sie Produkte und Services, die Ihre Kunden wirklich wollen, Campus Verlag

RIES (2014): Lean Startup - Schnell, risikolos und erfolgreich Unternehmen gründen, Verlag: Redline Verlag

weitere Literatur gemäß gesonderter Angabe in der VL

7 Informatik Bachelor - 7. Semester

7.1 Bachelor-Seminar

Name / engl.

Bachelor-Seminar / Bachelor Mentoring Workshop

Kürzel

BSEM

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Das Modul wird in deutscher und/oder englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.
Das Modul wird regelmäßig sowohl im Sommersemester als auch im Wintersemester angeboten.

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 5,
Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h

Lehrveranstaltungen

Bachelor-Seminar (4 SWS)

Lehr-/Lernmethoden

Seminar, Coaching, Praktikum, Übung

Prüfung

Prüfungsnummer

9051

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Portfolioprüfung

- Präsentation A, 15-30 Minuten, 20% (DVA-Seminar)
- Präsentation B, 15-30 Minuten, 30% (Bachelorseminar)
- Studienarbeit, 5 - 15 Seiten, 50% (DVA-Seminar)

Zusätzliche Informationen

hilfreiche Voraussetzungen

Das Bachelorseminar wird vorbereitend und begleitend zur Bachelorarbeit durchgeführt.

Inhalte des Moduls

Ziel ist es, die Studierenden vorbereitend und begleitend zur Bachelorarbeit an eine geeignete wissenschaftliche Methodik heranzuführen. Im Vordergrund steht die selbständige Erarbeitung von vertieften Kenntnissen sowie die aktive Beschäftigung im Rahmen von Einzelpräsentationen mit weiteren Beiträgen hierzu. Vorbereitend zur Bachelorarbeit arbeiten sich die Studierenden in einen abgeschlossenen Themenbereich ein und fassen Ihre Ergebnisse in einer Studienarbeit sowie in einer Präsentation(A) zusammen. Begleitend zur Bachelorarbeit werden Fragen, Problemstellungen und Lösungen mit dem betreuenden Professor diskutiert. Der Fortschritt bzw. die Ergebnisse der Bachelorarbeit sind in einer Präsentation(B) vorzustellen.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die Studierenden erlangen detaillierte Kenntnisse in den gewählten Themenbereichen sowie erweiterte Kompetenzen in den Bereichen Wissenschaftliches Arbeiten, Präsentation und Rhetorik.

Gewichtung der Einzelleistung in der Gesamtnote

Studienarbeit (50%), Präsentation(A) (20%), Präsentation(B) (30%)

Literaturliste

Es wird empfohlen mit Antritt des Seminars die Angebote der Hochschulbibliothek insbesondere zur „Recherche“ und „Zitieren“ zu nutzen. Hierzu können Sie die aktuellen Seminartermine auf folgender Webseite prüfen:

Seminare - Recherchieren, Wissenschaftliches Arbeiten, Zitieren und Literatur-/Wissensverwaltung: all das können Sie in unseren Bibliotheksseminaren an der Hochschule Augsburg lernen

7.2 Bachelorarbeit

Name / engl.

Bachelorarbeit / Bachelor Thesis

Kürzel

BA

Verantwortlicher

Professorinnen und Professoren der Fakultät für Informatik

Lehrsprache

Das Modul wird in deutscher und/oder englischer Sprache unterrichtet.

Fakultät

Fakultät für Informatik

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Das Modul wird regelmäßig sowohl im Wintersemester als auch im Sommersemester angeboten.

Arbeitsaufwand / Zusammensetzung

ECTS: 12, Arbeitsstunden: 360 h

Bearbeitungszeit: 4 Monate

Lehrveranstaltungen

Bachelorarbeit

Lehr-/Lernmethoden

Wissenschaftliches Arbeiten

Prüfung

Prüfungsnummer

9050

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Studienarbeit, 20-80 Seiten

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Bachelorarbeit können unter § 11 der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Literaturliste

Fachliteratur zur gewählten Fragestellung.

8 Wahlpflichtfächer

8.1 Allgemeinwissenschaftliches Wahlpflichtfach

Name / engl.

Allgemeinwissenschaftliches Wahlpflichtfach /

Kürzel

W.WAHL

Verantwortlicher

Studiengangsleiter

Lehrsprache

Das Modul wird in deutscher und englischer Sprache unterrichtet.

Fakultät

Fakultät für angewandte Geistes- und Naturwissenschaften

Verwendbarkeit

Informatik Bachelor

Dauer / Angebot

Die Dauer des Moduls beträgt ein Semester.

Das Modul wird regelmäßig sowohl im Sommersemester als auch im Wintersemester angeboten.

Arbeitsaufwand / Zusammensetzung

SWS: 4, CPs: 4,

Präsenzzeit: 60 h, Selbststudium: 60 h, Gesamtaufwand: 120 h

Lehrveranstaltungen

Als allgemeinwissenschaftliche Wahlpflichtfächer können alle an der Hochschule Augsburg angebotenen Lehrveranstaltungen gewählt werden, soweit sie nicht Pflicht- oder Wahlpflichtfächer dieses Studiengangs sind bzw. in der Ausschlussliste des Studiengangs geführt werden.

Lehr-/Lernmethoden

Die Lehr- und Lernmethoden, sowie die verwendeten Lehrmedien variieren je nach Veranstaltung.

Prüfung

Prüfungsnummer

Benotung

Gemäß § 20 der APO in der jeweils gültigen Fassung.

Prüfungsform

Die Art und Dauer der Prüfung variiert je nach Veranstaltung und ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

Zusätzliche Informationen

hilfreiche Voraussetzungen

Informationen zur Teilnahme an Wahlpflichtfächern können unter §7 der Studien- und Prüfungsordnung entnommen werden.

Inhalte des Moduls

Die allgemeinwissenschaftliche Ausbildung an der Hochschule Augsburg umfasst ein vielseitiges Angebot in geistes-, gesellschafts- und naturwissenschaftlichen Fächern. Die Studierenden lernen Wissensgebiete kennen, die über ihr fachspezifisches Studium hinausgehen.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Die allgemeinwissenschaftlichen Wahlpflichtfächer stellen gewissermaßen ein "Studium generale" dar. Die Studierenden sind in der Lage, das erworbene theoretische Wissen in Studium und Beruf praktisch anzuwenden.

Gewichtung der Einzelleistung in der Gesamtnote

Die Art und Dauer der Prüfung variiert je nach Veranstaltung und ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

Literaturliste

Die Literaturliste ist dem Modulhandbuch der Fakultät für angewandte Geistes- und Naturwissenschaften zu entnehmen.

8.2 Fachbezogene Wahlpflichtfächer

Name / engl.

Fachbezogene Wahlpflichtfächer / Optional required Subjects

Kürzel	Verantwortlicher Professorinnen und Professoren der Fakultät für Informatik
Lehrsprache Siehe Angaben des jeweiligen FWP-Fachs	Fakultät Fakultät für Informatik
Verwendbarkeit Informatik Bachelor	Dauer / Angebot Die Dauer des Moduls beträgt ein Semester. Das Modul wird sowohl im Sommersemester als auch im Wintersemester angeboten.
Arbeitsaufwand / Zusammensetzung SWS: 23, CPs: 25, Präsenzzeit: 345 h, Selbststudium: 405 h, Gesamtaufwand: 750 h	
Lehrveranstaltungen Die FWP-Fächer können aus dem Angebot der Fakultät für Informatik ausgewählt werden.	
Lehr-/Lernmethoden Seminaristischer Unterricht, Seminar, Praktikum, Directed Reading	
Prüfung	
Prüfungsnummer	Benotung Gemäß § 20 der APO in der jeweils gültigen Fassung.
Prüfungsform Weitere Informationen zu den fachbezogenen Wahlpflichtfächern finden Sie auf der Webseite des Studiengangs unter Studienrelevante Downloads.	
Zusätzliche Informationen	
hilfreiche Voraussetzungen Informationen zur Teilnahme an Wahlpflichtfächern können der Studien- und Prüfungsordnung entnommen werden.	

Inhalte des Moduls

Spezifische Fachkompetenz in den einzelnen Fächern.

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Spezifische Fachkompetenz in den jeweiligen Fächern.

Gewichtung der Einzelleistung in der Gesamtnote

Weitere Informationen zu den fachbezogenen Wahlpflichtfächern finden Sie auf der Webseite des Studiengangs unter Studienrelevante Downloads.

Literaturliste

Literatur wird im Seminar bekannt gegeben.

Index

Allgemeinwissenschaftliches
Wahlpflichtfach , 82

Bachelor-Seminar , 78

Bachelorarbeit , 80

Betriebssysteme , 54

Betriebswirtschaftslehre und DV-Recht ,
74

Datenbanken , 42

Datenkommunikation , 30

Englisch , 72

Fachbezogene Wahlpflichtfächer , 84

Grundlagen der Informatik 1 , 10

Grundlagen der Informatik 2 , 26

Mathematik 1 , 8

Mathematik 2 , 22

Numerische Mathematik , 46

Praktikum DVA , 66

Praktische Tätigkeit (Praxissemester) ,
58

Praxis-Seminar , 60

Programmieren 1 , 4

Programmieren 2 , 18

Programmieren 3 , 32

Projektarbeit 1 , 56

Projektarbeit 2 , 70

Rechnerstrukturen 1 , 14

Rechnerstrukturen 2 , 50

Software-Engineering 1 , 6

Software-Engineering 2 , 20

Software-Engineering 3 , 36

Statistik , 40

Systemnahe Programmierung , 62