



TECHNISCHE  
UNIVERSITÄT  
WIEN

Bachelor

Master

Doktorat

Universitäts-  
lehrgang

Studienplan (Curriculum)  
für das  
Bachelorstudium  
**Software & Information Engineering**  
**E 033 534**

Technische Universität Wien  
Beschluss des Senats der Technischen Universität Wien  
am 20. Juni 2022

Gültig ab 1. Oktober 2022

# Inhaltsverzeichnis

1. Grundlage und Geltungsbereich	3
2. Qualifikationsprofil	3
3. Dauer und Umfang	4
4. Zulassung zum Bachelorstudium	5
5. Aufbau des Studiums	5
6. Lehrveranstaltungen	14
7. Studieneingangs- und Orientierungsphase	14
8. Prüfungsordnung	16
9. Studierbarkeit und Mobilität	17
10. Bachelorarbeit	17
11. Akademischer Grad	17
12. Qualitätsmanagement	17
13. Inkrafttreten	20
14. Übergangsbestimmungen	20
A. Modulbeschreibungen	21
B. Lehrveranstaltungstypen	81
C. Zusammenfassung aller verpflichtenden Voraussetzungen	82
D. Semestereinteilung der Lehrveranstaltungen	83
E. Semesterempfehlung für schiefeinsteigende Studierende	85
F. Wahlfachkatalog „Transferable Skills“	89
G. Prüfungsfächer mit den zugeordneten Modulen und Lehrveranstaltungen	90
H. Bachelor-Abschluss with Honors	94

# 1. Grundlage und Geltungsbereich

Der vorliegende Studienplan definiert und regelt das ingenieurwissenschaftliche Bachelorstudium *Software & Information Engineering* an der Technischen Universität Wien. Es basiert auf dem Universitätsgesetz 2002 BGBl. I Nr. 120/2002 (UG) und dem Satzungsteil *Studienrechtliche Bestimmungen* der Technischen Universität Wien in der jeweils geltenden Fassung. Die Struktur und Ausgestaltung des Studiums orientieren sich an folgendem Qualifikationsprofil.

## 2. Qualifikationsprofil

Software Engineering beschäftigt sich mit der Entwicklung von Software von der Analyse über das Design und die Implementierung bis hin zur Qualitätssicherung und Wartung. Information Engineering beschäftigt sich mit der Erzeugung, Sammlung, Verarbeitung, Verteilung und Präsentation von Information. Beide Gebiete basieren auf wissenschaftlichen Prinzipien und Methoden.

Das Bachelorstudium *Software & Information Engineering* vermittelt eine breite, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Grundausbildung, welche die Absolvent\_innen sowohl für eine Weiterqualifizierung im Rahmen eines facheinschlägigen Masterstudiums als auch für eine Beschäftigung in beispielsweise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Entwicklung informationsverarbeitender Systeme sowohl als Expertin oder Experte im Team als auch in leitender Funktion
- Unterstützende Aufgaben in der Forschung

Aufgrund der beruflichen Anforderungen werden im Bachelorstudium *Software & Information Engineering* Qualifikationen hinsichtlich folgender Kategorien vermittelt.

**Fachliche und methodische Kompetenzen** Das Studium vermittelt grundlegende Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze sowie generell ein stabiles Grundlagen- und Methodenwissen vor allem in den folgenden Bereichen:

- Algorithmen und Datenstrukturen
- Architektur von Computer- und Softwaresystemen
- Mathematik, Wahrscheinlichkeitstheorie und Statistik
- Mensch-Maschine-Interaktion
- Programmierparadigmen
- Software Engineering
- Theoretische Informatik und Logik

Darüber hinaus werden die spezifischen Teilbereiche des Software und Information Engineering vermittelt:

- Betriebssysteme, Compiler

- Datenbanken, wissensbasierte Systeme
- Erhebung, Modellierung und Aufbereitung von Information
- Parallel Computing, Verteilte Systeme, Internet Computing
- Sicherheit und Recht
- Softwareerstellung und -wartung

**Kognitive und praktische Kompetenzen** Das Studium vermittelt generell wissenschaftlich fundierte Kompetenzen und die Fähigkeiten, auch neue Herausforderungen zu erkennen und kritisch zu hinterfragen sowie Probleme zu erkennen, zu formulieren, zu analysieren und zu lösen und deren Lösungen zu validieren. Durch die praktische Auseinandersetzung mit zukunftsorientierten Technologien, Methoden und Werkzeugen werden folgende kognitive und praktische Fertigkeiten vermittelt:

- Einsetzen formaler Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation
- Empirisch-experimentelle Systemvalidierung
- Entwicklung und Umsetzung von Design-Konzepten
- Interdisziplinäre und systemorientierte Denkweise
- Kritische Reflexion
- Methodisch fundierte Herangehensweise an Probleme, insbesondere im Umgang mit offenen/unspezifizierten Problemsituationen
- Präsentieren und Dokumentieren
- Umsetzen von Analyse-, Entwurfs-, Simulations- und Implementierungsstrategien
- Wissenschaftliches Arbeiten

**Soziale Kompetenzen und Selbstkompetenzen** Der Schwerpunkt liegt hier in der Vermittlung für Forschung und Beruf notwendiger sozialer Kompetenzen sowie auf der Förderung von Kreativitäts- und Innovationspotentialen.

- Aktive und passive Kritikfähigkeit
- Innovationsfähigkeit
- Kenntnisse der eigenen Fähigkeiten und Grenzen
- Neugierde, Eigeninitiative, Ausdauer, Flexibilität
- Reflexion der eigenen Arbeit und ihrer Wechselwirkung mit dem gesellschaftlichen, sozialen und beruflichen Kontext
- Selbstorganisation und Eigenverantwortlichkeit
- Teamfähigkeit
- Verantwortungsvoller Umgang mit Menschen sowie beruflichen und sozialen Gruppen in allen Tätigkeiten

### 3. Dauer und Umfang

Der Arbeitsaufwand für das Bachelorstudium *Software & Information Engineering* beträgt 180 ECTS-Punkte. Dies entspricht einer vorgesehenen Studiendauer von 6 Semestern als Vollzeitstudium.

ECTS-Punkte (ECTS) sind ein Maß für den Arbeitsaufwand der Studierenden. Ein Studienjahr umfasst 60 ECTS-Punkte, wobei ein ECTS-Punkt 25 Arbeitsstunden entspricht (gemäß § 54 Abs. 2 UG).

## 4. Zulassung zum Bachelorstudium

Voraussetzung für die Zulassung zum Bachelorstudium *Software & Information Engineering* ist die allgemeine Universitätsreife.

Personen, deren Erstsprache nicht Deutsch ist, haben die Kenntnis der deutschen Sprache, sofern dies gem. § 63 Abs. 1 Z 3 UG erforderlich ist, nachzuweisen.

In einzelnen Lehrveranstaltungen kann der Vortrag in englischer Sprache stattfinden bzw. können die Unterlagen in englischer Sprache vorliegen. Daher werden Englischkenntnisse auf Referenzniveau B1 des Gemeinsamen Europäischen Referenzrahmens für Sprachen empfohlen.

## 5. Aufbau des Studiums

Die Inhalte und Qualifikationen des Studiums werden durch *Module* vermittelt. Ein Modul ist eine Lehr- und Lerneinheit, welche durch Eingangs- und Ausgangsqualifikationen, Inhalt, Lehr- und Lernformen, den Regelarbeitsaufwand sowie die Leistungsbeurteilung gekennzeichnet ist. Die Absolvierung von Modulen erfolgt in Form einzelner oder mehrerer inhaltlich zusammenhängender *Lehrveranstaltungen*. Thematisch ähnliche Module werden zu *Prüfungsfächern* zusammengefasst, deren Bezeichnung samt Umfang und Gesamtnote auf dem Abschlusszeugnis ausgewiesen wird.

### Prüfungsfächer und zugehörige Module

Das Bachelorstudium *Software & Information Engineering* gliedert sich in nachstehende Prüfungsfächer mit den ihnen zugeordneten Modulen. Die mit Stern markierten Module sind *Wahl-*, die übrigen *Pflichtmodule*. Die Pflichtmodule sind in jedem Fall zu absolvieren. Aus der Liste der Wahlmodule sind Module in einem Gesamtumfang von mindestens 18 ECTS zu wählen. Im Rahmen des Moduls *Freie Wahlfächer und Transferable Skills* sind so viele Lehrveranstaltungen zu absolvieren, dass ihr Umfang zusammen mit den ECTS-Punkten der Lehrveranstaltungen aus den Pflichtmodulen und dem Umfang der gewählten Wahlmodule mindestens 180 ECTS ergibt. Werden in den Wahlmodulen insgesamt mehr als 18 ECTS absolviert, können im Modul *Freie Wahlfächer und Transferable Skills* im gleichen Ausmaß weniger ECTS absolviert werden, jedoch sind darin mindestens 6 ECTS aus dem Bereich der Transferable Skills zu absolvieren.

### Algorithmen und Programmierung

Algorithmen und Datenstrukturen (8,0 ECTS)

Einführung in die Programmierung (9,5 ECTS)

Einführung in paralleles Rechnen (Parallel Computing) (6,0 ECTS)  
Programmierparadigmen (6,0 ECTS)

- \*Deklaratives Problemlösen (6,0 ECTS)
- \*Funktionale Programmierung (6,0 ECTS)
- \*Logikprogrammierung und Constraints (6,0 ECTS)

### **Computersysteme**

Betriebssysteme (6,0 ECTS)  
Einführung in Visual Computing (6,0 ECTS)  
Grundzüge digitaler Systeme (6,0 ECTS)  
Verteilte Systeme (6,0 ECTS)

- \*Abstrakte Maschinen (6,0 ECTS)
- \*Übersetzerbau (6,0 ECTS)
- \*Grundlagen der Computergraphik (6,0 ECTS)
- \*Zuverlässige Echtzeitsysteme (6,0 ECTS)

### **Informatik und Gesellschaft**

Denkweisen der Informatik (6,5 ECTS)  
Interface und Interaction Design (6,0 ECTS)  
Einführung in Security (6,0 ECTS)  
Daten- und Informatikrecht (mindestens 6,0 ECTS)

- \*Security (mindestens 6,0 ECTS)

### **Information Engineering**

Datenbanksysteme (6,0 ECTS)  
Einführung in Artificial Intelligence (6,0 ECTS)

- \*Einführung in wissensbasierte Systeme (6,0 ECTS)
- \*Logik für Wissensrepräsentation (6,0 ECTS)
- \*Semistrukturierte Daten (6,0 ECTS)
- \*Statistische Datenanalyse (mindestens 6,0 ECTS)
- \*Web Engineering (6,0 ECTS)

### **Mathematik, Statistik und Theoretische Informatik**

Algebra und Diskrete Mathematik (9,0 ECTS)  
Analysis (6,0 ECTS)  
Statistik und Wahrscheinlichkeitstheorie (6,0 ECTS)  
Theoretische Informatik (6,0 ECTS)

- \*Argumentieren und Beweisen (6,0 ECTS)
- \*Introduction to Numerics (4,5 ECTS)
- \*Logic and Reasoning in Computer Science (6,0 ECTS)
- \*Multivariate und computerintensive statistische Methoden (9,0 ECTS)

### **Software Engineering**

Software Engineering (6,0 ECTS)  
 Software Engineering Projekt (6,0 ECTS)

- \*Programm- und Systemverifikation (6,0 ECTS)
- \*Softwareprojekt-Beobachtung und -Controlling (6,0 ECTS)
- \*Software-Qualitätssicherung (6,0 ECTS)
- \*Usability Engineering and Mobile Interaction (6,0 ECTS)

### **Freie Wahlfächer und Transferable Skills**

Freie Wahlfächer und Transferable Skills (18,0 ECTS)

### **Bachelorarbeit**

Bachelorarbeit (13,0 ECTS)

### **Kurzbeschreibung der Module**

Dieser Abschnitt charakterisiert die Module des Bachelorstudiums *Software & Information Engineering* in Kürze. Eine ausführliche Beschreibung ist in Anhang A zu finden.

**Abstrakte Maschinen (6,0 ECTS)** Dieses Modul vermittelt die theoretischen Grundlagen und konkrete Ausprägungen von abstrakten Maschinen. Dazu gehören Grundlagen über die effiziente Implementierung von abstrakten Maschinen und konkrete Maschinen wie die Java Virtual Machine, die Dalvik Virtual Machine, die Warren Abstract Machine und die SECD Maschine. Praktische Fertigkeiten werden durch die Implementierung einer eigenen abstrakten Maschine im Übungsteil erworben. Einfache Kenntnisse aus dem Übersetzerbau werden vorausgesetzt.

**Algebra und Diskrete Mathematik (9,0 ECTS)** Das Modul vermittelt zentrale Grundlagenkenntnisse, Theoreme und Beweistechniken der Algebra (algebraische Strukturen und lineare Algebra) und der Diskreten Mathematik (Kombinatorik und Graphentheorie). Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen. Neben der Vertiefung des Verständnisses und der Vernetzung der Vorlesungsinhalte dient der Übungsteil vor allem der Entwicklung von praktischen Fertigkeiten in der Erstellung korrekter mathematischer Beweise sowie in der mathematischen Modellierung und Analyse von Anwendungsproblemen.

**Algorithmen und Datenstrukturen (8,0 ECTS)** Dieses Modul führt Studierende in grundsätzliche Methoden zur Entwicklung und Analyse von Algorithmen ein. Neben

Fachkenntnissen zu fundamentalen Algorithmen und Datenstrukturen erwerben sich die Studierenden die Fähigkeit zum Einsatz theoretisch fundierter Methoden zur Analyse von Algorithmen. Eine abstrakte und effizienzorientierte Denkweise wird gefördert.

**Analysis (6,0 ECTS)** Das Modul vermittelt zentrale Grundlagenkenntnisse, Theoreme und Beweistechniken in der mathematischen Analysis (Folgen und Reihen, elementare Funktionen, Differential- und Integralrechnung in einer Variablen). Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen. Neben der Vertiefung des Verständnisses und der Vernetzung der Vorlesungsinhalte dient der Übungsteil vor allem der Entwicklung von praktischen Fertigkeiten zur Erstellung korrekter mathematischer Beweise sowie zur mathematischen Modellierung und Analyse von Anwendungsproblemen.

**Argumentieren und Beweisen (6,0 ECTS)** Das Modul bietet eine Einführung in die zentralen Beweistechniken. Es setzt sich aus einem Vorlesungsteil und einem begleitenden Übungsteil zusammen, welcher der Vertiefung der Vorlesungsinhalte und der Entwicklung von Fertigkeiten zur Erstellung korrekter mathematischer Beweise dient. Schwerpunkte sind die Strukturierung von Beweisen und Argumentationen sowie die unterschiedlichen Techniken zur Induktion, die an praktischen Fragestellungen der Informatik demonstriert werden.

**Bachelorarbeit (13,0 ECTS)** Ein Seminar führt in die wissenschaftliche Methodik und in den Wissenschaftsbetrieb ein. Darauf aufbauend bearbeitet der oder die Studierende im Rahmen eines Projektes ein dem Qualifikationsprofil des Studiums entsprechendes Thema und beschreibt Aufgabenstellung, Methodik, Umfeld und Ergebnisse in einer schriftlichen Bachelorarbeit. Das Thema der Bachelorarbeit wird auf dem Abschlusszeugnis ausgewiesen.

**Betriebssysteme (6,0 ECTS)** Dieses Modul vermittelt grundlegende Kenntnisse über Betriebssysteme, deren Architektur, Funktionsweise und wesentliche Komponenten. Die Grundkonzepte und theoretischen Inhalte werden in der Vorlesung, das Arbeiten mit Betriebssystemen und Betriebssystemmechanismen zusätzlich in praktischen Übungen im Labor vermittelt.

Vorausgesetzt werden Kenntnisse der Grundlagen der Informatik sowie Programmierkenntnisse.

**Datenbanksysteme (6,0 ECTS)** Das Modul vermittelt Grundkenntnisse von Datenmodellierung und Datenbankmanagementsystemen. Es bildet die Basis für die Verwendung von Datenbanksystemen bei künftigen Aufgaben im Bereich Softwareentwicklung. Der Schwerpunkt liegt auf dem relationalen Datenmodell. Neben den grundlegenden Techniken der Datenmodellierung wird die Umsetzung in ein relationales Schema sowie die Verwendung einer relationalen Datenbank vermittelt. Außerdem werden Kenntnisse über zentrale Datenbankkonzepte wie Transaktionen, Fehlerbehandlung/Recovery und Mehrbenutzersynchronisation vermittelt.

**Deklaratives Problemlösen (6,0 ECTS)** In diesem Modul werden vertiefende Kenntnisse zur Lösung komplexer computationaler Probleme mittels deklarativer Techniken

vermittelt. Die Teilnehmer erlernen theoretische und anwendungsorientierte Aspekte unterschiedlicher Werkzeuge, welche auf klassischer Logik und Logikprogrammierung basieren.

**Denkweisen der Informatik (6,5 ECTS)** Studierende werden mit einer Reihe verschiedener Denkweisen und Denkmodelle konfrontiert, die unterschiedliche Herangehensweisen an Probleme implizieren. Darüber hinaus lernen Studierende ausgewählte Aspekte der Geschichte der Informatik kennen, reflektieren die Rolle der Informatik in der Gesellschaft, und setzen sich exemplarisch mit besonderen Fragen aus diesem Bereich auseinander. Schließlich bietet das Modul einen Überblick und eine Einführung in die Themen des wissenschaftlichen Arbeitens sowie zum Lernen und Arbeiten an der TU Wien. Darüber hinaus gibt das Modul einen Überblick über die Informatikstudien, die Forschungsgebiete der Informatik und die Organisation von Fakultät und Universität, und vermittelt die Verhaltensregeln der Informatik sowie Strategien für einen erfolgreichen Studienabschluss.

**Einführung in Artificial Intelligence (6,0 ECTS)** Studierende mit Kenntnissen in Datenstrukturen und Algorithmen sowie mit Fertigkeiten in der Mathematik (wie z.B. Beweise selbst zu führen) erhalten in diesem Modul (a) grundlegende Kenntnisse der theoretischen Grundlagen und Methoden der Artificial Intelligence (AI) und (b) fundamentale Konzepte die zum Verständnis der Arbeitsweise als auch zur Erstellung von AI Systemen notwendig sind.

**Einführung in die Programmierung (9,5 ECTS)** Das Modul richtet sich an ProgrammieranfängerInnen und bildet die Basis für die weitere Programmierausbildung. Der Schwerpunkt liegt auf einer systematischen Vorgehensweise beim Programmieren. Studierende erwerben neben Fachkenntnissen vor allem praktische Fertigkeiten in der Programmierung. Abstrakte Denkweisen werden gefördert.

**Einführung in paralleles Rechnen (Parallel Computing) (6,0 ECTS)** Das Modul gibt eine Einführung in das parallele Rechnen auf unterschiedlichen Rechnerarchitekturen, vom Mehrkern-Prozessor bis zum Hochleistungsrechensystem. Es werden Lösungsstrategien für spezifische Probleme und konkrete Programmierschnittstellen aufgezeigt.

**Einführung in Security (6,0 ECTS)** IT-Sicherheit ist ein kritisches Element erfolgreicher IT-Projekte. Trotz funktional gut ausgeführter Projekte können diese bei schweren Sicherheitsproblemen je nach Anwendungsgebiet geschäftsschädigende Auswirkungen haben. In diesem Modul lernen die Studierenden Sicherheitsprobleme zu erkennen und Sicherheitsmaßnahmen anzuwenden, um IT-Projekte auch aus Sicherheitssicht erfolgreich abzuschließen.

**Einführung in Visual Computing (6,0 ECTS)** Das Modul Einführung in Visual Computing vermittelt einen Überblick über die Aufgaben und Problemstellungen sowie die Methoden des Visual Computing, und ein kritisches Verständnis ihrer Theorien und Grundsätze. Der Begriff Visual Computing ist durch das methodische Zusammenwachsen der Bereiche Bildverarbeitung, Computer Vision, Computergraphik, Visualisierung und Mensch-Maschine-Interaktion entstanden, und umfasst außer diesen Themen auch

Bereiche wie Augmented und Virtual Reality und maschinelles Lernen. Um dieses Modul absolvieren zu können werden Grundkenntnisse im Programmieren und solide Mathematikkenntnisse (Maturaniveau + Mathematik 1) vorausgesetzt.

**Einführung in wissensbasierte Systeme (6,0 ECTS)** Studierende mit elementaren Logikkenntnissen sowie mit Fertigkeiten in der Mathematik (wie z.B. Beweise selbst zu führen) erhalten in diesem Modul (a) grundlegende Kenntnisse in den theoretischen Grundlagen wissensbasierter Systeme und (b) fundamentale Konzepte, die zum Verständnis der Arbeitsweise als auch zur Erstellung wissensbasierter Systeme notwendig sind.

**Freie Wahlfächer und Transferable Skills (18,0 ECTS)** Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

**Funktionale Programmierung (6,0 ECTS)** Das Modul vertieft die Konzepte des funktionalen Programmierparadigmas, seiner theoretischen Grundlagen und der entsprechenden Sprachkonstrukte und Programmier Techniken funktionaler Sprachen und funktionaler Programmierung. Einen Schwerpunkt bildet dabei das praktische Lösen von Programmieraufgaben in einer fortgeschrittenen modernen funktionalen Programmiersprache.

**Grundlagen der Computergraphik (6,0 ECTS)** Das Modul vermittelt tiefergehende Kenntnisse im Bereich Computergraphik, aufbauend auf das Modul "Einführung in Visual Computing". Die Inhalte werden theoretisch durch eine Vorlesung und praktisch durch eine Übung vermittelt.

**Grundzüge digitaler Systeme (6,0 ECTS)** Das Modul vermittelt die Grundkenntnisse für die formale Modellierung statischer und dynamischer Systeme sowie für das Verständnis und den Entwurf digitaler Systeme.

**Interface und Interaction Design (6,0 ECTS)** Interface and Interaction Design beschäftigt sich mit Fragen der Gestaltung Interaktiver Systeme unter Berücksichtigung der Komplexität realweltlicher Kontexte. Im Mittelpunkt stehen dabei die Theorien, methodischen Vorgehensweisen und Praktiken der Inkludierung diverser und widersprüchlicher Anforderungen und Bedürfnisse von Menschen und Situationen.

**Introduction to Numerics (4,5 ECTS)** Studierende werden mit den grundlegenden Konzepten algorithmisch-numerischer Lösungsmethoden vertraut gemacht. Inhaltlich gehören dazu grundlegende Fehlerbegriffe (Datenfehler, Verfahrens- oder Diskretisierungsfehler, Rundungsfehler), Kondition mathematischer Probleme, numerische Lösung linearer und nichtlinearer Gleichungssysteme, polynomiale Interpolation und Approximation, numerische Integration, numerische Lösungen von Differentialgleichungen, Design und Verwendung numerischer Algorithmen bzw. numerischer Software.

**Logic and Reasoning in Computer Science (6,0 ECTS)** Aufbauend auf elementaren Kenntnissen (wie Automaten und Aussagenlogik), vermittelt dieses Modul (i) die

logischen Grundlagen der Informatik; (ii) die Fähigkeit, formal-mathematische Beschreibungen verfassen zu können; (iii) und computergestützte Begründungen und Argumentationsmethoden zu entwerfen.

**Logik für Wissensrepräsentation (6,0 ECTS)** Dieses Modul vermittelt grundlegende Kenntnisse unterschiedlicher Logiken bzw. logikbasierter Formalismen zur Wissensrepräsentation. Beginnend mit der klassischen Logik als zentraler Wissensrepräsentations-sprache werden unterschiedliche Formalismen zum Nichtmonotonen Schließen, zum para-konsistenten Schließen, sowie zur Wissensrevision (Belief Revision) sowie der Modallogik vorgestellt. Weiters werden grundlegende Aspekte ontologischen Modellierens diskutiert.

**Logikprogrammierung und Constraints (6,0 ECTS)** Dieses Modul vermittelt das logikorientierte Programmierparadigma anhand praktischer Aufgaben. Als Grundpro-grammiersprache wird ISO-Prolog verwendet. Schwerpunkt ist der pure und monotone Teil der Sprache. Darauf aufbauend kommen in ISO-Prolog eingebettete Constraint-Programmiersprachen zum Einsatz.

**Multivariate und computerintensive statistische Methoden (9,0 ECTS)** Dieses Modul vermittelt multivariate statistische Methoden und computerintensive Methodiken der Statistik mittels statistischer Simulation.

**Programmierparadigmen (6,0 ECTS)** Das Modul veranschaulicht Konzepte, Techni-ken und Denkweisen von Programmierparadigmen, insbesondere des objektorientierten, funktionalen, nebenläufigen (concurrent) und parallelen Paradigmas. Im Vordergrund steht das praktische Lösen von Programmieraufgaben auf für die Paradigmen typische Weise.

**Programm- und Systemverifikation (6,0 ECTS)** Das Modul bietet eine Einfüh- rung in Methoden zur computerunterstützten Verifikation und Qualitätssicherung von Software und Hardware. Die in der Vorlesung vermittelten Grundlagen und Methoden werden an Hand von theoretischen und praktischen Aufgabenstellungen vertieft und in geeigneten Anwendungen erprobt.

**Daten- und Informatikrecht (mindestens 6,0 ECTS)** Dieses Modul eröffnet den Zugang zu den für das Internet bzw. die Informationsgesellschaft relevanten rechtlichen Aspekten und sensibilisiert für aktuelle rechtspolitische Problemstellungen. Ferner lei- stet es einen Beitrag zur Reduktion der rechtlichen Risiken, denen Techniker\_innen im Rahmen ihrer beruflichen Praxis ausgesetzt sind.

**Security (mindestens 6,0 ECTS)** IT-Sicherheit ist ein kritisches Element erfolgrei- cher IT-Projekte. Trotz funktional gut ausgeführter Projekte können diese bei schweren Sicherheitsproblemen je nach Anwendungsgebiet geschäftsschädigende Auswirkungen ha- ben. In den Lehrveranstaltungen dieses Moduls lernen die Studierenden Sicherheitspro- bleme zu erkennen und Sicherheitsmaßnahmen anzuwenden, um IT-Projekte auch aus Sicherheitssicht erfolgreich abzuschließen.

**Semistrukturierte Daten (6,0 ECTS)** Das Modul *Semistrukturierte Daten* behandelt die Erstellung, Verarbeitung und den Zugriff auf Daten, die nicht so hochgradig struktu- riert sind, wie in traditionellen relationalen Datenbanksystemen. Das Modul behandelt

einige Prinzipien von schwach- bzw. semi-strukturierten Daten (z.B. die Verwendung von Bäumen und Graphen anstatt von Datenbanktabellen), einige existierende Standards um solche Daten zu erfassen und zu teilen, einige Standards um die Integrität solcher Daten sicherzustellen, sowie Standards und Technologien, um Abfragen über schwach strukturierten Daten auszuwerten bzw. schwach strukturierte Daten in Applikationen zu nutzen.

**Software Engineering (6,0 ECTS)** Das Modul vermittelt Studierenden mit Grundkenntnissen in der individuellen Programmierung grundlegende Kenntnisse und Fertigkeiten zur Softwareerstellung und Wartung durch das Zusammenführen der isolierten Kenntnisse und Fähigkeiten aus den relevanten vorangehenden Lehrveranstaltungen zu einer praxisnahen Gesamtsicht von der softwaretechnischen Problemstellung bis zur Lösung. Der Schwerpunkt liegt auf einer systematischen Vorgehensweise für die Softwareentwicklung von mittelgroßen Softwareprodukten in einem Team mit klar definierten Rollen.

**Software Engineering Projekt (6,0 ECTS)** In diesem Modul wenden die Studierenden Software Engineering Methoden auf eine reale Aufgabenstellung an. Der Schwerpunkt liegt auf einer systematischen Vorgehensweise für die Softwareentwicklung von mittelgroßen Softwareprodukten in einem Team mit klar definierten Rollen.

**Softwareprojekt-Beobachtung und -Controlling (6,0 ECTS)** Das Vertiefungsmodul *Softwareprojekt-Beobachtung und -Controlling* vermittelt eine Sichtweise von IT- und Software-Projekten, die oberhalb des klassischen Projektmanagements in der Softwareentwicklung und unterhalb des sehr übergreifenden IT-Controllings als gesamtbetriebliche Instanz liegt. Auf Basis praxisnaher Beispiele werden Methoden zur Überwachung und Steuerung komplexer Projektsituationen präsentiert. Das Lehrziel sind Fertigkeiten aus folgenden Themengebieten:

- Identifikation von Kennziffern für das Benchmarking und Evaluieren eines Projektportfolios
- Nutzbringende Darstellung der Dimensionen Zeit, Kosten und Qualitätsmetriken der SW Herstellung
- Supervision: Überwachung und Eskalationsmechanismus
- Strategie: Analyse und Behebung von Fehlstellungen in der SW Erzeugung; SE Theorie vs. SE Praxis
- Vergleichbarkeit: Controlling über mehrere SW Projekte innerhalb eines Unternehmens

**Software-Qualitätssicherung (6,0 ECTS)** Dieses Modul vermittelt Studierenden mit Grundkenntnissen in der Software-Entwicklung eine Einführung in formale und angewandte Kenntnisse, Methoden und Kompetenzen zur Beurteilung und Verbesserung der Qualität von Softwaresystemen im wissenschaftlichen und industriellen Umfeld. Der Schwerpunkt liegt auf einer systematischen Vorgehensweise für Reviews und Testen von Artefakten aus der Entwicklung von Softwaresystemen, die aus mehreren Komponenten bestehen.

**Grundzüge digitaler Systeme (6,0 ECTS)** Das Modul vermittelt die Grundkenntnisse für die formale Modellierung statischer und dynamischer Systeme sowie für das Verständnis und den Entwurf digitaler Systeme.

**Statistik und Wahrscheinlichkeitstheorie (6,0 ECTS)** Das Modul vermittelt Grundkenntnisse der Wahrscheinlichkeitstheorie und Statistik.

**Statistische Datenanalyse (mindestens 6,0 ECTS)** Dieses Modul vermittelt Grundkenntnisse der statistischen Datenanalyse und der computerorientierten Statistik. Mit Computerstatistik wird ein Zugang zu statistischen Methoden vermittelt, der sowohl formal orientiert ist, bei dem aber auch anhand von konkreten Problemstellungen die theoretischen Konzepte praxisnah mit dem Computer gelöst werden.

**Theoretische Informatik (6,0 ECTS)** Aufbauend auf elementaren Kenntnissen formaler Modellierungssprachen (wie Automaten oder Aussagenlogik) zur Spezifikation realer Sachverhalte vermittelt dieses Modul die theoretischen und logischen Grundlagen der Informatik und die Fähigkeit, formal-mathematische Beschreibungen verstehen und verfassen zu können.

**Übersetzerbau (6,0 ECTS)** Das Modul vermittelt die theoretischen Grundlagen des Übersetzerbaus und die praktischen Fähigkeiten der Entwicklung von Parsern und Übersetzern. Es werden alle Phasen eines Übersetzers von der lexikalischen Analyse, der Syntaxanalyse, der semantischen Analyse, der Optimierung und der Codeerzeugung abgedeckt. Weiters wird noch auf die Implementierung von objektorientierten Programmiersprachen eingegangen. In Vorlesungen werden die theoretischen Grundlagen vermittelt, in einer Laborübung in geführten Kleingruppen werden die Inhalte in Form von Programmieraufgaben praktisch geübt.

**Usability Engineering and Mobile Interaction (6,0 ECTS)** Dieses Modul vermittelt theoretische Grundlagen und praktische Methoden in den Bereichen Usability Engineering und User-Centered Interaction Research für mobile Anwendungen. Im Bereich Usability Engineering stehen die Qualitätskriterien für gute User Interfaces und die Methoden zu deren Evaluierung im Vordergrund. Aufbauend darauf wird im zweiten Teil des Moduls der Fokus auf den mobilen Bereich gelegt und dessen Besonderheiten anhand von Fallbeispielen hervorgehoben. Die in der Vorlesung vermittelten Kenntnisse sollen in den jeweiligen Übungsblöcken in Kleingruppen praktisch erprobt werden.

**Verteilte Systeme (6,0 ECTS)** Das Modul *Verteilte Systeme* vermittelt maßgebliche Konzepte verteilter Systeme sowie aktuelle Entwicklungen in diesem Bereich. Daher wird die Rolle verteilter Systeme in aktuellen Systemlandschaften diskutiert. Weiterhin werden Anforderungen an (große) verteilte Systeme und verschiedene Arten von verteilten Systemen vorgestellt. Der Fokus liegt auf fundamentalen Konzepten, Methoden und Algorithmen für verteilte Systeme, sowie deren Vor- und Nachteile und Einsatzmöglichkeiten. Ziel der Übung ist das Erlernen von grundlegenden Techniken wie beispielsweise Sockets, Remote Method Invocations (RMI), sowie einfachen Sicherheits-Mechanismen in verteilten Systemen. Die Übung verleiht in diesem Zusammenhang praxisnahe Fähigkeiten in der Netzwerk-Programmierung sowie beim Entwickeln von verteilten Anwendungen.

**Web Engineering (6,0 ECTS)** Das Modul *Web Engineering* beschäftigt sich mit der systematischen Entwicklung von dynamischen Web-Anwendungen im Intra- und Internet mit Technologien und unter Berücksichtigung geltender Standards, u.a. für Barrierefreiheit.

**Zuverlässige Echtzeitsysteme (6,0 ECTS)** Das Modul vermittelt die wesentlichen Kenntnisse für die Spezifikation, den Entwurf, die Implementierung und das Testen von fehlertoleranten, sowie sicherheitskritischen verteilten Echtzeitsystemen. Fehlerarten, Fehlermodellierung, Fehlermaskierung, der Umgang mit zeitabhängiger Information, die Konstruktion von Computersystemen mit strikten Anforderungen im Zeitbereich und die Auswirkungen dieser Faktoren auf die Sicherheit von Computersystemen sind dabei zentrale Aspekte. Die Grundlagen zu zuverlässigen Systemen und Echtzeitsystemen werden in Vorlesungen vermittelt. Problemstellungen aus der Simulation von Fehlertoleranten Systemen und der Fehleranalyse/modellierung werden in praktischen Übungen behandelt.

## 6. Lehrveranstaltungen

Die Stoffgebiete der Module werden durch Lehrveranstaltungen vermittelt. Die Lehrveranstaltungen der einzelnen Module sind in Anhang A in den jeweiligen Modulbeschreibungen spezifiziert. Lehrveranstaltungen werden durch Prüfungen im Sinne des Universitätsgesetzes beurteilt. Die Arten der Lehrveranstaltungsbeurteilungen sind in der Prüfungsordnung (Abschnitt 8) festgelegt.

Betreffend die Möglichkeiten der Studienkommission, Module um Lehrveranstaltungen für ein Semester zu erweitern, und des Studienrechtlichen Organs, Lehrveranstaltungen individuell für einzelne Studierende Wahlmodulen zuzuordnen, wird auf § 27 des Studierenrechtlichen Teils der Satzung der TU Wien verwiesen.

## 7. Studieneingangs- und Orientierungsphase

Die Studieneingangs- und Orientierungsphase (StEOP) soll den Studierenden eine verlässliche Überprüfung ihrer Studienwahl ermöglichen. Sie leitet vom schulischen Lernen zum universitären Wissenserwerb über und schafft das Bewusstsein für die erforderliche Begabung und die nötige Leistungsbereitschaft.

Die Studieneingangs- und Orientierungsphase des Bachelorstudiums *Software & Information Engineering* gilt als positiv absolviert, wenn die Lehrveranstaltung

5,5 VU Einführung in die Programmierung 1

sowie Lehrveranstaltungen im Ausmaß von 11,0 ECTS aus den Modulen

*Algebra und Diskrete Mathematik, Analysis, Datenbanksysteme, Denkweisen der Informatik* und *Grundzüge digitaler Systeme*

und aus diesen jedenfalls die Lehrveranstaltung

1,0 VU Orientierung Informatik und Wirtschaftsinformatik

sowie eine der Lehrveranstaltungen

4,0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
und

9,0 VU Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
positiv abgeschlossen wurden.

Vor positiver Absolvierung der StEOP dürfen weitere Lehrveranstaltungen im Umfang von 22 ECTS absolviert werden, die aus den folgenden Lehrveranstaltungen gewählt werden können:

8,0 VU Algorithmen und Datenstrukturen

4,0 VU Einführung in die Programmierung 2

6,0 VU Einführung in Visual Computing

Weiters können Lehrveranstaltungen im Rahmen des Moduls *Freie Wahlfächer und Transferable Skills* gewählt werden, sofern deren Absolvierung nicht anderweitig beschränkt ist.

Die positiv absolvierte Studieneingangs- und Orientierungsphase ist jedenfalls Voraussetzung für die Absolvierung der im Bachelorstudium vorgesehenen Lehrveranstaltungen, in deren Rahmen die Bachelorarbeit abzufassen ist.

## **Wiederholbarkeit von Teilleistungen**

Für alle StEOP-Lehrveranstaltungen müssen mindestens zwei Antritte im laufenden Semester vorgesehen werden, wobei einer der beiden auch während der lehrveranstaltungs-freien Zeit abgehalten werden kann. Es muss ein regulärer, vollständiger Besuch der Vorträge mit prüfungsrelevanten Stoff im Vorfeld des ersten Prüfungstermins möglich sein.

Bei Lehrveranstaltungen mit einem einzigen Prüfungsakt ist dafür zu sorgen, dass die Beurteilung des ersten Termins zwei Wochen vor dem zweiten Termin abgeschlossen ist, um den Studierenden, die beim ersten Termin nicht bestehen, ausreichend Zeit zur Einsichtnahme in die Prüfung und zur Vorbereitung auf den zweiten Termin zu geben.

Die Beurteilung des zweiten Termins ist vor Beginn der Anmeldung für prüfungsimmanente Lehrveranstaltungen des Folgesemesters abzuschließen.

Bei prüfungsimmanenten Lehrveranstaltungen ist dies sinngemäß so anzuwenden, dass entweder eine komplette Wiederholung der Lehrveranstaltung in geblockter Form angeboten wird oder die Wiederholbarkeit innerhalb der Lehrveranstaltung sichergestellt wird.

Wiederholbarkeit innerhalb der Lehrveranstaltung bedeutet, dass Teilleistungen, ohne die keine Beurteilung mit einem Notengrad besser als „genügend“ (4) bzw. „mit Erfolg teilgenommen“ erreichbar ist, jeweils wiederholbar sind. Teilleistungen sind Leistungen, die gemeinsam die Gesamtnote ergeben und deren Beurteilungen nicht voneinander abhängen. Diese Wiederholungen zählen nicht im Sinne von § 15 (6) des studienrechtlichen Teils der Satzung der TU Wien als Wiederholung.

Zusätzlich können Gesamtprüfungen angeboten werden, wobei eine derartige Gesamtprüfung wie ein Prüfungstermin für eine Vorlesung abgehalten werden muss.

## 8. Prüfungsordnung

Für den Abschluss des Bachelorstudiums ist die positive Absolvierung der im Studienplan vorgeschriebenen Module erforderlich. Ein Modul gilt als positiv absolviert, wenn die ihm zuzurechnenden Lehrveranstaltungen gemäß Modulbeschreibung positiv absolviert wurden.

Das Abschlusszeugnis beinhaltet

- (a) die Prüfungsfächer mit ihrem jeweiligen Umfang in ECTS-Punkten und ihren Noten,
- (b) das Thema der Bachelorarbeit und
- (c) die Gesamtbeurteilung sowie
- (d) auf Antrag des\_der Studierenden die Gesamtnote des absolvierten Studiums gemäß §72a UG.

Die Note eines Prüfungsfaches ergibt sich durch Mittelung der Noten jener Lehrveranstaltungen, die dem Prüfungsfach über die darin enthaltenen Module zuzuordnen sind, wobei die Noten mit dem ECTS-Umfang der Lehrveranstaltungen gewichtet werden. Bei einem Nachkommateil kleiner gleich 0,5 wird abgerundet, andernfalls wird aufgerundet. Wenn keines der Prüfungsfächer schlechter als mit „gut“ und mindestens die Hälfte mit „sehr gut“ benotet wurde, so lautet die *Gesamtbeurteilung* „mit Auszeichnung bestanden“ und ansonsten „bestanden“.

Die Studieneingangs- und Orientierungsphase gilt als positiv absolviert, wenn die im Studienplan vorgegebenen Leistungen zu Absolvierung der StEOP erbracht wurden.

Lehrveranstaltungen des Typs VO (Vorlesung) werden aufgrund einer abschließenden mündlichen und/oder schriftlichen Prüfung beurteilt. Alle anderen Lehrveranstaltungen besitzen immanenten Prüfungscharakter, d.h., die Beurteilung erfolgt laufend durch eine begleitende Erfolgskontrolle sowie optional durch eine zusätzliche abschließende Teilprüfung.

Zusätzlich können zur Erhöhung der Studierbarkeit Gesamtprüfungen zu prüfungsimmanenten Lehrveranstaltungen angeboten werden, wobei diese wie ein Prüfungstermin für eine Vorlesung abgehalten werden müssen und § 15 (6) des Studienrechtlichen Teils der Satzung der TU Wien hier nicht anwendbar ist.

Der positive Erfolg von Prüfungen und wissenschaftlichen sowie künstlerischen Arbeiten ist mit „sehr gut“ (1), „gut“ (2), „befriedigend“ (3) oder „genügend“ (4), der negative Erfolg ist mit „nicht genügend“ (5) zu beurteilen. Bei Lehrveranstaltungen, bei denen eine Beurteilung in der oben genannten Form nicht möglich ist, werden diese durch „mit Erfolg teilgenommen“ (E) bzw. „ohne Erfolg teilgenommen“ (O) beurteilt.

Die Beurteilung der Lehrveranstaltung

1,0 VU Orientierung Informatik und Wirtschaftsinformatik

erfolgt bei positivem Erfolg durch „mit Erfolg teilgenommen“, andernfalls durch „ohne Erfolg teilgenommen“; sie bleibt bei der Berechnung der gemittelten Note des Prüfungsfaches unberücksichtigt.

## 9. Studierbarkeit und Mobilität

Studierende des Bachelorstudiums *Software & Information Engineering*, die ihre Studienwahl im Bewusstsein der erforderlichen Begabungen und der nötigen Leistungsbereitschaft getroffen und die Studieneingangs- und Orientierungsphase, die dieses Bewusstsein vermittelt, absolviert haben, sollen ihr Studium mit angemessenem Aufwand in der dafür vorgesehenen Zeit abschließen können.

Den Studierenden wird empfohlen, ihr Studium nach dem Semestervorschlag in Anhang D zu absolvieren. Studierenden, die ihr Studium im Sommersemester beginnen, wird empfohlen, ihr Studium nach der Semesterempfehlung in Anhang E zu absolvieren.

Die Beurteilungs- und Anwesenheitsmodalitäten von Lehrveranstaltungen der Typen UE, LU, PR, VU, SE und EX sind im Rahmen der Lehrvereinbarungen mit dem Studienrechtlichen Organ festzulegen und den Studierenden in geeigneter Form, zumindest in der elektronisch zugänglichen Lehrveranstaltungsbeschreibung anzukündigen, soweit sie nicht im Studienplan festgelegt sind. Für mindestens eine versäumte oder negative Teilleistung, die an einem einzigen Tag zu absolvieren ist (z.B. Test, Klausur, Laborübung), ist zumindest ein Ersatztermin spätestens innerhalb von 2 Monaten anzubieten.

Die Anerkennung von im Ausland absolvierten Studienleistungen erfolgt durch das studienrechtliche Organ. Zur Erleichterung der Mobilität stehen die in § 27 Abs. 1 bis 3 der *Studienrechtlichen Bestimmungen* der Satzung der Technischen Universität Wien angeführten Möglichkeiten zur Verfügung. Diese Bestimmungen können in Einzelfällen auch zur Verbesserung der Studierbarkeit eingesetzt werden.

Die Zahl der jeweils verfügbaren Plätze und das Verfahren zur Vergabe dieser Plätze in Lehrveranstaltungen mit beschränkten Ressourcen wird von der Lehrveranstaltungsleitung festgelegt und vorab bekannt gegeben. Die Lehrveranstaltungsleitung ist berechtigt, für ihre Lehrveranstaltung Ausnahmen von der Teilnahmebeschränkung zuzulassen.

## 10. Bachelorarbeit

Die Bachelorarbeit ist eine im Bachelorstudium eigens anzufertigende schriftliche Arbeit, welche eigenständige Leistungen beinhaltet. Sie besitzt einen Regelarbeitsaufwand von 10 ECTS und kann im Rahmen des Moduls *Bachelorarbeit* erstellt werden.

## 11. Akademischer Grad

Den Absolvent\_innen des Bachelorstudiums *Software & Information Engineering* wird der akademische Grad *Bachelor of Science* – abgekürzt *BSc* – verliehen.

## 12. Qualitätsmanagement

Das Qualitätsmanagement des Bachelorstudiums *Software & Information Engineering* gewährleistet, dass das Studium in Bezug auf die studienbezogenen Qualitätsziele der

TU Wien konsistent konzipiert ist und effizient und effektiv abgewickelt sowie regelmäßig überprüft wird. Das Qualitätsmanagement des Studiums erfolgt entsprechend des Plan-Do-Check-Act Modells nach standardisierten Prozessen und ist zielgruppenorientiert gestaltet. Die Zielgruppen des Qualitätsmanagements sind universitätsintern die Studierenden und die Lehrenden sowie extern die Gesellschaft, die Wirtschaft und die Verwaltung, einschließlich des Arbeitsmarktes für die Studienabgänger\_innen.

In Anbetracht der definierten Zielgruppen werden sechs Ziele für die Qualität der Studien an der TU Wien festgelegt: (1) In Hinblick auf die Qualität und auf die Aktualität des Studienplans ist die Relevanz des Qualifikationsprofils für die Gesellschaft und den Arbeitsmarkt gewährleistet. In Hinblick auf die Qualität der inhaltlichen Umsetzung des Studienplans sind (2) die Lernergebnisse in den Modulen des Studienplans geeignet gestaltet um das Qualifikationsprofil umzusetzen, (3) die Lernaktivitäten und -methoden geeignet gewählt um die Lernergebnisse zu erreichen und (4) die Leistungsnachweise geeignet um die Erreichung der Lernergebnisse zu überprüfen. (5) In Hinblick auf die Studierbarkeit der Studienpläne sind die Rahmenbedingungen gegeben um diese zu gewährleisten. (6) In Hinblick auf die Lehrbarkeit verfügt das Lehrpersonal über fachliche und zeitliche Ressourcen um qualitätsvolle Lehre zu gewährleisten.

Um die Qualität der Studien zu gewährleisten, werden der Fortschritt bei Planung, Entwicklung und Sicherung aller sechs Qualitätsziele getrennt erhoben und publiziert. Die Qualitätssicherung überprüft die Erreichung der sechs Qualitätsziele. Zur Messung des ersten und zweiten Qualitätszieles wird von der Studienkommission zumindest einmal pro Funktionsperiode eine Überprüfung des Qualifikationsprofils und der Modulbeschreibungen vorgenommen. Zur Überprüfung der Qualitätsziele zwei bis fünf liefert die laufende Bewertung durch Studierende, ebenso wie individuelle Rückmeldungen zum Studienbetrieb an das Studienrechtliche Organ, laufend ein Gesamtbild über die Abwicklung des Studienplans. Die laufende Überprüfung dient auch der Identifikation kritischer Lehrveranstaltungen, für welche in Abstimmung zwischen Studienrechtlichem Organ, Studienkommission und Lehrveranstaltungsleiter\_innen geeignete Anpassungsmaßnahmen abgeleitet und umgesetzt werden. Das sechste Qualitätsziel wird durch qualitätssichernde Instrumente im Personalbereich abgedeckt. Zusätzlich zur internen Qualitätssicherung wird alle sieben Jahre eine externe Evaluierung der Studien vorgenommen.

Jedes Modul besitzt eine\_n Modulverantwortliche\_n. Diese Person ist für die inhaltliche Kohärenz und die Qualität der dem Modul zugeordneten Lehrveranstaltungen verantwortlich. Diese wird insbesondere durch zyklische Kontrollen, inhaltliche Feinabstimmung mit vorausgehenden und nachfolgenden Modulen sowie durch Vergleich mit analogen Lehrveranstaltungen bzw. Modulen anderer Universitäten im In- und Ausland sichergestellt.

## **Lehrveranstaltungskapazitäten und Teilnahmebeschränkungen**

Für die verschiedenen Typen von Lehrveranstaltungen (siehe Anhang B) dienen die folgenden Gruppengrößen als Richtwert:

Lehrveranstaltungstyp	Gruppengröße	
	je Leiter(in)	je Tutor(in)
VO	200	
UE mit Tutor(inn)en	50	20
UE	20	
LU mit Tutor(inn)en	40	15
LU	15	
EX, PR, SE	20	

Für Lehrveranstaltungen des Typs VU werden für den Vorlesungs- bzw. Übungsteil die Gruppengrößen für VO bzw. UE herangezogen. Die Beauftragung der Lehrenden erfolgt entsprechend der tatsächlichen Abhaltung.

Zur Gewährleistung der Studierbarkeit gemäß § 58 Abs.7 und 8 UG werden in allen Lehrveranstaltungen Studierende, die zum Bachelorstudium emphSoftware & Information Engineering zugelassen sind und diese Lehrveranstaltungen im Rahmen ihres Studiums verpflichtend zu absolvieren haben, bevorzugt aufgenommen. Die Anmeldung Studierender anderer Studien zu den Lehrveranstaltungen (außer vom Typ VO) sowie die Prüfungsberechtigung in Lehrveranstaltungen des Typs VO des Bachelorstudiums *Software & Information Engineering* setzt die bereits erfolgreich absolvierte STEOP im jeweiligen eigenen Studium voraus.

Lehrveranstaltungen mit ressourcenbedingten Teilnahmebeschränkungen sind in der Beschreibung des jeweiligen Moduls entsprechend gekennzeichnet; weiters sind dort die Anzahl der verfügbaren Plätze und das Verfahren zur Vergabe dieser Plätze festgelegt. Die Lehrveranstaltungsleiter\_innen sind berechtigt, mehr Teilnehmer\_innen zu einer Lehrveranstaltung zuzulassen als nach Teilnahmebeschränkungen oder Gruppengrößen vorgesehen, sofern dadurch die Qualität der Lehre nicht beeinträchtigt wird.

Kommt es in einer Lehrveranstaltung ohne explizit geregelte Platzvergabe zu einem unvorhergesehenen Andrang, kann die Lehrveranstaltungsleitung in Absprache mit dem studienrechtlichen Organ Teilnahmebeschränkungen vornehmen und die Vergabe der Plätze nach folgenden Kriterien (mit absteigender Priorität) regeln.

- Es werden jene Studierenden bevorzugt aufgenommen, die die formalen und inhaltlichen Voraussetzungen erfüllen. Die inhaltlichen Voraussetzungen können etwa an Hand von bereits abgelegten Prüfungen oder durch einen Eingangstest überprüft werden.
- Unter diesen hat die Verwendung der Lehrveranstaltung als Pflichtfach Vorrang vor der Verwendung als Wahlfach und diese vor der Verwendung als Freifach.
- Innerhalb dieser drei Gruppen sind jeweils jene Studierenden zu bevorzugen, die trotz Vorliegens aller Voraussetzungen bereits in einem früheren Abhaltesemester abgewiesen wurden.

Die Studierenden sind darüber ehebaldigst zu informieren.

## **13. Inkrafttreten**

Dieser Studienplan tritt mit 1. Oktober 2022 in Kraft.

## **14. Übergangsbestimmungen**

Die Übergangsbestimmungen werden gesondert im Mitteilungsblatt verlautbart und liegen im Dekanat der Fakultät für Informatik auf.

## A. Modulbeschreibungen

Die den Modulen zugeordneten Lehrveranstaltungen werden in folgender Form angeführt:

9,9/9,9 XX Titel der Lehrveranstaltung

Dabei bezeichnet die erste Zahl den Umfang der Lehrveranstaltung in ECTS-Punkten und die zweite ihren Umfang in Semesterstunden. ECTS-Punkte sind ein Maß für den Arbeitsaufwand der Studierenden, wobei ein Studienjahr 60 ECTS-Punkte umfasst und ein ECTS-Punkt 25 Stunden zu je 60 Minuten entspricht. Eine Semesterstunde entspricht so vielen Unterrichtseinheiten wie das Semester Unterrichtswochen umfasst. Eine Unterrichtseinheit dauert 45 Minuten. Der Typ der Lehrveranstaltung (XX) ist in Anhang *Lehrveranstaltungstypen* auf Seite 81 im Detail erläutert.

### Abstrakte Maschinen

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- alle theoretischen Grundlagen von abstrakten Maschinen verstehen und
- Details konkreter abstrakter Maschinen erklären.

*Kognitive und praktische Kompetenzen:* Die Auseinandersetzung mit konkreten Beispielen von abstrakten Maschinen und die Implementierung eigener abstrakter Maschinen ermöglicht die Studierenden

- die Qualität von abstrakten Maschinen zu beurteilen,
- eigene abstrakte Maschinen zu entwerfen und
- abstrakte Maschinen zu implementieren.

*Soziale Kompetenzen und Selbstkompetenzen:* Eigeninitiative und Neugierde auf innovative und kreative Konzepte und Lösungsansätze werden besonders gefördert.

**Inhalt:**

- reale Maschinen, Prozessorarchitekturen
- Interpretationstechniken (threaded code), Implementierung von Forth
- Pascal P4 Maschine
- Java Virtuelle Maschine (just-in-time Übersetzung), Microsoft Intermediate Language
- Registermaschinen und die DalvikVM
- syntaxgesteuerte Editoren und Baummaschinen
- Prologmaschinen (WAM, VAM)
- funktionale Maschinen (Lamda Kalkül, SECD Maschine)

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Grundlagen von Programmiersprachen und Übersetzerbau

*Kognitive und praktische Kompetenzen:* Programmierkenntnisse

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Einführung in die Programmierung, Programmierparadigmen, Übersetzerbau.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Vortrag und selbständiges Erlernen der eher theoretischen Grundlagen. Die Beurteilung erfolgt durch Prüfung. Übung am Computer zur Entwicklung praktischer Fähigkeiten zur Entwicklung von abstrakten Maschinen. Die Leistungsbeurteilung erfolgt durch die Beurteilung der Implementierung einer selbst entworfenen abstrakten Maschine und der Präsentation dieser Implementierung.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Abstrakte Maschinen

**Algebra und Diskrete Mathematik**

**Regelarbeitsaufwand:** 9,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Reproduzieren bzw. Herleiten der wichtigsten mathematischen Definitionen, Theoreme und Beweismethoden der Algebra und Diskreten Mathematik.

*Kognitive und praktische Kompetenzen:* Finden von Beweisen für mathematische Problemstellungen aus Algebra und Diskreter Mathematik; Modellieren einfacher Anwendungsprobleme aus Informatik, Naturwissenschaften und Technik als mathematische Problemstellungen und Lösen derselben mit geeigneten mathematischen Methoden.

*Soziale Kompetenzen und Selbstkompetenzen:* Präsentieren von Problemlösungen vor einer Übungsgruppe.

**Inhalt:**

- Grundlagen: elementare Logik (Aussagen, Implikation, Kontraposition, Verneinung, Quantoren); elementare Beweistechniken (direkter und indirekter Beweis, Gegenbeispiele); elementare Zahlentheorie.
- Mengenlehre: Grundlagen (Venn-Diagramme, Komplemente, kartesisches Produkt, Potenzmenge); Funktionen (Mengenrelationen, surjektive, injektive, bijektive Funktionen, Komposition); Relationen (Äquivalenzrelation, Partitionen, Ordnungsrelation, Maximumsprinzip); Kardinalität und Abzählbarkeit (endliche, unendlichen und abzählbare Mengen).
- Induktion: Induktionsprinzip (vollständige Ind., transfinite Ind.); rekursive Definitionen.

- Grundlagen der Kombinatorik: Abzählprinzipien (Summen- und Produktregel); Schubfachschluss; Inklusions-Exklusions-Prinzip; kombinatorische Grundaufgaben (Permutationen, Auswahlen, Partitionen); elementare Identitäten (Binomischer Lehrsatz, binomische Identitäten); Rekursionen (Fibonacci-Zahlen, Derangements, Turm von Hanoi); Lösungsmethoden für Rekursionen (Rekursionen erster Ordnung, lineare Rekursionen mit konstanten Koeffizienten).
- Graphentheorie: Grundlagen (gerichtete, ungerichtete, bipartite Graphen, Wege, etc.); Handshake-Lemma; Eulersche und Hamiltonsche Linien; Graphrelationen (Isomorphie, Subgraphen, Minore); Zusammenhang (Zusammenhangskomponenten, Menger's theorem); azyklische Graphen; ebene Graphen (inkl. Eulersche Polyederformel); elementare Graph-Algorithmen (Azyklizität, Kruskal-Alg., minimaler Spannbaum, Dijkstra-Alg.).
- Algebraische Strukturen: Gruppentheorie (inkl. Faktorgruppen, Homomorphiesatz, zyklische Gruppen, direkte Produkte); Ringe (Integritätsbereiche, Ideale); Körper (Polynomringe über Körpern); Verbände.
- Lineare Algebra: Vektoren; Matrizen; lineare Abbildungen; lineare Gleichungssysteme; Determinanten; Eigenwerte und Eigenvektoren; Skalarprodukte, Orthogonalität.
- Grundlagen algebraische Codierungstheorie: Gruppencodes, Linearcodes.

**Erwartete Vorkenntnisse:** Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Wöchentliche Vorlesung mit kontinuierlicher begleitender Übung (individuell auszuarbeitende Übungsbeispiele, Lösungspräsentation an der Tafel), wodurch die in der Vorlesung vermittelten Inhalte effizient erlernt und die mathematische Problemlösungskompetenz trainiert wird. Leistungsfeststellung durch mehrere Lösungspräsentationen, Übungstests, Abschlussprüfung.

**Lehrveranstaltungen des Moduls:**

4,0/4,0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

5,0/2,0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

oder

9,0/6,0 VU Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

# Algorithmen und Datenstrukturen

**Regelarbeitsaufwand:** 8,0 ECTS

## Lernergebnisse:

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können Studierende Folgendes beschreiben:

- fundamentale Algorithmen und Datenstrukturen,
- Methoden zur Bewertung und Analyse von Algorithmen, und
- eine systematische Vorgehensweise zur Entwicklung von Algorithmen.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- abstrakt und effizienzorientiert an die Entwicklung von Algorithmen herangehen,
- theoretisch fundierte Methoden zur Analyse von Algorithmen benutzen, und
- ihre Kenntnisse von fundamentalen Algorithmen und Datenstrukturen anwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- selbstorganisiert und eigenverantwortlich effiziente Lösungsansätze entwickeln und
- die eigenen Lösungsansätze präsentieren.

## Inhalt:

- Fundamentale Prinzipien der Algorithmenanalyse
- Asymptotische Schranken für Laufzeit und Speicherplatzbedarf
- Fundamentale Datenstrukturen (z.B. Listen, Graphen, Suchbäume)
- Fundamentale algorithmische Prinzipien (z.B. Greedy, Divide-and-Conquer, Branch-and-Bound, Approximation, Dynamische Programmierung, Lokale Suche, Hashing)
- Problemlösungsstrategien und Optimierung
- Handhabbarkeit, Polynomialzeitreduktionen, NP-Vollständigkeit

**Erwartete Vorkenntnisse:** Inhalte der LVA *Einführung in die Programmierung 1* sowie fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Übungsaufgaben bestehen vorwiegend aus Aufgaben die schriftlich ausgearbeitet werden. Sie werden örtlich ungebunden innerhalb vorgegebener Fristen gelöst, die Lösungen werden in Übungsgruppen vorgestellt. Die Beurteilung erfolgt auf Basis mehrerer schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen.

## Lehrveranstaltungen des Moduls:

8,0/5,5 VU Algorithmen und Datenstrukturen

# Analysis

**Regelarbeitsaufwand:** 6,0 ECTS

## Lernergebnisse:

*Fachliche und methodische Kompetenzen:* Reproduzieren bzw. Herleiten der wichtigsten mathematischen Definitionen, Theoreme und Beweismethoden der mathematischen Analysis.

*Kognitive und praktische Kompetenzen:* Finden von Beweisen für mathematische Problemstellungen aus der Analysis; Modellieren einfacher Anwendungsprobleme aus Informatik, Naturwissenschaften und Technik als mathematische Problemstellungen und Lösen derselben mit geeigneten Verfahren zur analytischen und numerischen Problemlösung.

*Soziale Kompetenzen und Selbstkompetenzen:* Präsentieren von Problemlösungen vor einer Übungsgruppe.

## Inhalt:

- Folgen, Reihen und Funktionen: Folgen reeller Zahlen (Grenzwert, Monotonie und Beschränktheit, Konvergenzuntersuchungen); unendliche Reihen (Konvergenzkriterien, Cauchyprodukt und Potenzreihen); asymptotischer Vergleich von Folgen (Landausymbole:  $O()$ ,  $o()$ ,  $\Omega()$ ).
- Elementare Funktionen: Potenzen mit reellen Exponenten; Exponentialfunktion und Logarithmus; Darstellung der Exponentialfunktion; Winkelfunktionen und Arcusfunktionen.
- Grenzwerte und Nullstellen von Funktionen, Stetigkeit: metrische und topologische Grundbegriffe (offene, geschlossene Mengen, Umgebungen, Basis, Häufungspunkte); Umgebungs- und Folgenstetigkeit Eigenschaften stetiger Funktionen: Nullstellensatz, Zwischenwertsatz, Monotonie.
- Differentialrechnung in einer Variablen: Differenzenquotient und Differenzierbarkeit; Ableitung einfacher Funktionen; Eigenschaften und Ableitungsregeln; Mittelwertsatz der Differentialrechnung; Taylorreihen; Monotonie und die erste Ableitung; höhere Ableitungen; verallgemeinerter Mittelwertsatz und die Regel von de l'Hospital.
- Integralrechnung in einer Variablen: Definition und Eigenschaften Riemann-Integral; Integration als Umkehrung der Differentiation, Fläche unter Kurven; Techniken des Integrierens; Mittelwert- und Hauptsatz der Differential- und Integralrechnung; uneigentliche Integrale.
- Elementare Differentialgleichungen: lineare Differentialgleichungen erster Ordnung.
- Grundlagen Differentialrechnung in mehreren Variablen: Funktionen in mehreren Variablen; partielle Ableitungen, totale Ableitung; Ableitungsregeln; Richtungsableitung; Taylorentwicklung; Hauptsatz über implizite Funktionen; lokale Extrema.

- Computer-Numerik: Zahlendarstellungsfehler; Konversionsfehler; Fehlerfortpflanzung (Summe, Produkte, Polynome, elementare Funktionen); algorithmische Fehlerfortpflanzung, Konditionszahlen.

**Erwartete Vorkenntnisse:** Fundierte Mathematik-Kenntnisse auf AHS/BHS-Maturaniveau.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Wöchentliche Vorlesungen mit kontinuierlich begleitender Übung (individuell auszuarbeitende Übungsbeispiele, Lösungspräsentation an der Tafel), wodurch die in der Vorlesung vermittelten Inhalte effizient erlernt und die mathematische Problemlösungskompetenz trainiert wird. Leistungsfeststellung durch mehrere Lösungspräsentationen, Übungstests, Abschlussprüfung.

**Lehrveranstaltungen des Moduls:**

2,0/2,0 VO Analysis für Informatik und Wirtschaftsinformatik

4,0/2,0 UE Analysis für Informatik und Wirtschaftsinformatik

oder

6,0/4,0 VU Analysis für Informatik und Wirtschaftsinformatik

## Argumentieren und Beweisen

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden die wesentlichen mathematischen Schlussweisen und Beweistechniken benennen, die Korrektheit der Schlussweisen argumentieren und den Zusammenhang der Beweistechniken mit Kalkülen der formalen Logik herstellen.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden die Korrektheit gegebener Beweise argumentieren, (auch komplexere) Beweise selbst erstellen und strukturieren, unterschiedliche Induktionsprinzipien korrekt anwenden, sowie Induktionshypothesen kreativ erstellen.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden die Beweisideen und Beweise kommunizieren.

**Inhalt:**

- Was ist ein Beweis? Welche Aufgaben hat er?
- Einfache Beweistechniken
- Beweis von All- und Existenzaussagen, Konjunktionen, Disjunktionen, Implikationen, Äquivalenzen
- Nutzung dieser Aussagen in einem Beweis

- Zusammenhang zum Kalkül des natürlichen Schliessens
- Was ist Induktion? Wozu wird sie benötigt?
- Arten der Induktion (mathematische, starke, strukturelle, Noether'sche), jeweils mit Diskussion des entsprechenden Induktionsschemas und Anwendungsfälle (ausführlich demonstriert an Beispielen)
- Wie schreibt man einen Induktionsbeweis?

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls die grundlegenden Beweisprinzipien benennen und beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls einfache natürlichsprachlich gegebene Sachverhalte korrekt formalisieren und diese beweisen können. Desweiteren sollen die Studierenden vor der Absolvierung des Moduls einfache Programmieraufgaben als rekursives Programm formulieren können.

*Soziale Kompetenzen und Selbstkompetenzen:* Keine speziellen Voraussetzungen.

Diese Voraussetzungen werden im Modul *Algebra und Diskrete Mathematik* vermittelt.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Geblockte Einführungsvorlesung (im Gesamtumfang von knapp 1,5 ECTS), danach umfangreiche individuell auszuarbeitende Aufgaben zum Argumentieren und Beweisen (im Umfang von 4,5 ECTS). Ausführliche Präsentation der Beweise (alle Lösungen durch jede Teilnehmerin/jeden Teilnehmer). Exemplarische Ausarbeitung einiger Lösungen, Korrektur durch LVA Leiter/Tutoren zwecks Rückmeldung. Leistungsermittlung auf Grund der Präsentationen und der berichtigten schriftlichen Ausarbeitungen.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Argumentieren und Beweisen

## **Bachelorarbeit**

**Regelarbeitsaufwand:** 13,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Wissenschaftliche Methodik
- Internationaler Wissenschaftsbetrieb

*Kognitive und praktische Kompetenzen:*

- Systematische Recherche
- Präsentationstechniken

- Strukturierte und konzise Kommunikation von Inhalten in mündlicher und schriftlicher Form
- Fähigkeit zur Anwendung der im Studium erworbenen Kenntnisse und Fertigkeiten im Kontext einer größeren Problemstellung

*Soziale Kompetenzen und Selbstkompetenzen:*

- Selbstorganisation
- Eigenverantwortlichkeit und Eigeninitiative
- Teamfähigkeit
- Finden kreativer Problemlösungen
- Reflexion der eigenen Arbeit im technischen und gesellschaftlichen Kontext

**Inhalt:** Im Rahmen des Seminars *Wissenschaftliches Arbeiten* lernen die Studierenden wissenschaftliche Methoden und den Wissenschaftsbetrieb kennen. An Hand eines vorgegebenen Themas üben sie Recherche sowie schriftliche und mündliche Präsentation. Darauf aufbauend wenden sie im Projekt *Bachelorarbeit für Informatik und Wirtschaftsinformatik* die im Studium erworbenen Kenntnisse und Fertigkeiten auf ein Thema an, das dem Qualifikationsprofil des Studiums entspricht. Die erzielten Ergebnisse werden neben der Aufgabenstellung, den angewandten Methoden und dem Umfeld in einer schriftlichen Abschlussarbeit dargestellt.

**Erwartete Vorkenntnisse:** Die Arbeit an der Bachelorarbeit erfordert die Kenntnisse, Fertigkeiten und Kompetenzen zumindest der Pflichtmodule des Bachelorstudiums.

**Verpflichtende Voraussetzungen:** Positive Absolvierung der Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Im Seminar besteht bei den Vorträgen zu Wissenschaftsmethodik und -betrieb sowie bei der Präsentation der Rechercheergebnisse Anwesenheitspflicht, ebenso bei der Präsentation der Bachelorarbeiten. Davon abgesehen können das Seminar- und das Bachelorarbeitsthema in Absprache mit den Lehrenden zeitlich und örtlich weitgehend ungebunden bearbeitet werden. Die Beurteilung orientiert sich an der Qualität und Originalität der mündlichen und schriftlichen Darstellung der Themen sowie der dafür notwendigen Vorarbeiten und berücksichtigt auch das Engagement bei der Diskussion der Arbeiten anderer Studierender.

**Lehrveranstaltungen des Moduls:**

10,0/5,0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik

3,0/2,0 SE Wissenschaftliches Arbeiten

## **Betriebssysteme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Die Studierenden verstehen den Aufbau und die Funktionsweise von Computersystemen bzw. Prozessoren und wie Betriebssysteme die Ressourcen dieser

Systeme - Rechenzeit, Speicher, Dateien und I/O-Geräte - verwalten. Designentscheidungen und Trade-Offs bei der Realisierung von Betriebssystemen können dargelegt werden. Weiters können Studierende nach Absolvierung der Lehrveranstaltung erklären, welche Mechanismen und Systemdatenstrukturen für die parallele (nebenläufige) Abarbeitung von Prozessen notwendig sind, und sind in der Lage, parallele Prozesse und deren koordinierten Ablauf zu programmieren.

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Aufbau und Funktionsweise von Prozessoren und deren wichtigsten Komponenten sowie deren Funktion und Zusammenspiel skizzieren,
- die Rolle und Aufgaben von Betriebssystemen erklären,
- Designentscheidungen für Managementmechanismen von Systemressourcen diskutieren bzw. aus gegebenen Anforderungen ableiten,
- Mechanismen zur Koordination und Synchronisation paralleler Prozesse verstehen und Koordinations- und Synchronisationsaufgaben mit diesen Mechanismen lösen,
- Prinzipien und Mechanismen des Zugriffsschutzes beschreiben.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Programmieraufgaben unter Verwendung des Application Programming Interfaces (API) eines Betriebssystems lösen und Betriebssystemservices über diese Programmierschnittstelle nutzen,
- gemeinsame Ressourcen und Kommunikations- sowie Synchronisationsmechanismen eines Betriebssystems zur Programmierung paralleler Prozesse verwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Entwicklungen von Betriebssystemtechnologien diskutieren und bewerten,
- Abstraktionen ableiten,
- Probleme des Ressourcenmanagements und Synchronisationsaufgaben lösen.

### **Inhalt:**

- Überblick Architektur und Arbeitsweise von Computersystemen bzw. Prozessoren
- Grundkonzepte Betriebssysteme
- Prozesse, Threads und Scheduling
- Prozesssynchronisation und Deadlock
- Speicherverwaltung
- Ein/Ausgabe und Disk Management
- Security und Protection
- Arbeiten mit Betriebssystemen

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Kenntnisse von Zahlendarstellungen in Computern, der grundlegenden Funktionsweise von Computern, endlicher Automaten, Transducer, Grammatiken, Programmiersprachen, sowie Kenntnisse der systematischen Vorgehensweise bei der Programmerstellung.

*Kognitive und praktische Kompetenzen:* Interpretieren und Arbeiten mit Zahlendarstellungen und Automaten. Kenntnisse der Programmierung in einer Programmiersprache und der systematischen Programmerstellung und Evaluation.

*Soziale Kompetenzen und Selbstkompetenzen:* Analyse komplexer Zusammenhänge und Wechselwirkungen, Strukturieren und Entwerfen von modularen, interagierenden Systemen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Einführung in die Programmierung, Grundzüge digitaler Systeme.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Das Modul setzt sich aus einem Vorlesungsteil und einem Laborübungsteil zusammen. Eine Einführung zu Mikroprozessorsystemarchitekturen, sowie die Grundlagen, zentralen Konzepte und theoretischen Inhalte zu Betriebssystemen werden im Vorlesungsteil präsentiert. Ausgewählte Inhalte und Problemstellungen aus dem Bereich der Betriebssystemprogrammierung werden im Labor unter UNIX (Linux) programmiert. Einführungswissen zu den zu lösenden Aufgabenstellungen wird in begleitenden Vortragsblöcken angeboten. Schwerpunkte des Laborteils sind:

- Arbeiten unter Unix/Linux: Shell, Prozesse, Signale, Filesystem
- Programmieren mit der Systemprogrammiersprache C, Debugging
- Systemprogrammierung mit folgenden Mechanismen
  - Parameter und Optionsbehandlung, Filebehandlung
  - Sockets
  - Signale und Signalbehandlung
  - verwandte Prozesse (fork, exec, wait)
  - Kommunikationsmechanismen: Named und Unnamed Pipes, Message Queues
  - Synchronisation mit Semaphoren bzw. Sequencer und Eventcounts
  - Kommunikation über Shared Memory
  - Ressourcenverwaltung

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Betriebssysteme

## **Datenbanksysteme**

**Regelarbeitsaufwand:** 6,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls kön-

nen die Studierenden die unter „Inhalt“ angeführten Konzepte und Techniken mit fachspezifischer Terminologie beschreiben.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- Datenmodelle mittels ER- und EER-Diagrammen erstellen,
- EER-Diagramme in ein relationales Schema in 3. Normalform umsetzen,
- SQL für die Manipulation und Abfrage von Daten verwenden,
- einfache Anfragen in relationaler Algebra und Relationenkalkül verstehen und selbst formulieren,
- Programmieraufgaben mit einer prozeduralen Datenbankprogrammiersprache lösen,
- unterschiedliche Isolations-Levels im Mehrbenutzerbetrieb gezielt einsetzen.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- gestellte Aufgaben selbständig und fristgerecht lösen,
- die erstellten Lösungen kommunizieren und begründen,
- ein deklaratives Programmierparadigma (SQL) anwenden.

#### **Inhalt:**

- Datenbankentwurf, Datenmodellierung mittels ER- und EER-Diagrammen,
- relationales Datenmodell,
- Umsetzung eines EER-Diagramms in ein relationales Schema in dritter Normalform,
- funktionale Abhängigkeiten, Normalformen,
- relationale Abfragesprachen (relationale Algebra, Relationenkalkül, SQL),
- komplexe Schemadefinitionen (Constraints, Views),
- komplexe SQL Abfragen (Schachtelung, Rekursion),
- prozedurale Datenbankprogrammierung,
- Transaktionen,
- Fehlerbehandlung/Recovery,
- Mehrbenutzersynchronisation.

#### **Erwartete Vorkenntnisse:**

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls

- mathematische Notationen lesen und schreiben können,
- grundlegende Datenstrukturen und Algorithmen verwenden können,
- eine allgemeine imperative Programmiersprache anwenden können,
- grundlegende Formalismen der Modellierung anwenden können,

- grundlegende Begriffe und Konzepte der Logik (Aussagenlogik, Prädikatenlogik) beschreiben und anwenden können.

Diese Vorkenntnisse werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Einführung in die Programmierung.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden in Vorträgen vorgestellt und in begleitenden Übungen von den Studierenden erarbeitet. Die Beurteilung erfolgt auf Basis schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen. Der Übungsbetrieb und die Tests können computerunterstützt durchgeführt werden.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Datenbanksysteme

## Deklaratives Problemlösen

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden unterschiedliche Werkzeuge, Sprachen und logikorientierte Programmiermethoden zum deklarativen Problemlösen benennen und erläutern, sowie theoretische Zusammenhänge korrekt argumentieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- die eingesetzten Techniken und Methoden formal analysieren,
- Methoden und Techniken für eine vorgegebene Aufgabenstellung zielgerichtet auswählen, sowie
- Lösungen und Formalismen kritisch bewerten.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden erarbeitete Lösungen kommunizieren.

**Inhalt:**

- Grundlagen moderner Entscheidungsprozeduren für die Erfüllbarkeit aussagenlogischer und quantifizierter aussagenlogischer Formeln (SAT und QSAT Solver)
- Normalformtransformationen
- Problemlösen mittels SAT und QSAT Solver
- Systeme und Semantiken der Logikprogrammierung
- Eigenschaften der Antwortmengenprogrammierung
- Praktische Anwendungen der Antwortmengenprogrammierung zur Lösung computationaler Probleme

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls die wesentlichen Konzepte der Aussagen- und Prädikatenlogik erster Stufe benennen und beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls

- Deduktionskonzepte und Beweisprinzipien zur Erstellung eigener Beweise korrekt anwenden,
- die Korrektheit der einzelnen Beweisschritte formal argumentieren, sowie
- vorgegebene Problemstellungen algorithmisch umsetzen können.

*Soziale Kompetenzen und Selbstkompetenzen:* Keine speziellen Voraussetzungen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Algorithmen und Datenstrukturen, Einführung in Artificial Intelligence, Grundzüge digitaler Systeme, Theoretische Informatik.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Lehrveranstaltung des Moduls besteht aus einem Vorlesungsteil (Frontalvortrag) und einem begleitenden Übungsteil. Die Vorlesung dient zur Vermittlung der theoretischen Grundlagen des besprochenen Fachgebietes während in der Übung die Teilnehmer in selbständiger Weise Lösungen zu konkreten Aufgabenstellungen erarbeiten. Die Beurteilung der Vorlesung erfolgt auf Basis von Prüfungen (schriftlich und/oder mündlich) und die Beurteilung der Übung anhand der abgegebenen Lösungen der Aufgabenstellungen sowie mittels Abgabegesprächen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Deklaratives Problemlösen

## **Denkweisen der Informatik**

**Regelarbeitsaufwand:** 6,5 ECTS

**Lernergebnisse:** Denkweisen der Informatik bietet eine Einführung und einen Überblick über die Informatik aus der Sicht ihrer Arbeits- und Denkweisen, vermittelt als eine Art angewandter Wissenschaftstheorie. Die LVA soll Interesse am weiteren Studium wecken, und die Studierenden in die Lage versetzen, die im weiteren Studium präsentierte Inhalte besser einzuordnen. Die Studierenden sollen so in die Lage versetzt werden, die Informatik sowohl als Wissenschaft als auch als Praxis nachhaltiger zu verstehen, und dieses Wissen im Rahmen des Studiums produktiv umzusetzen.

*Fachliche und methodische Kompetenzen:* Studierende können ...

- erklären, was Informatik ist;
- die Strukturen und Prozesse einer Universität darstellen;

- Lernmethoden und Organisationsformen für das erfolgreiche Fortkommen im eigenen Studium anwenden;
- ableiten, dass es bei Problemformulierung und Problemlösung unterschiedliche und zum Teil in Konflikt zueinander stehende Sichtweisen, Herangehensweisen und Motive gibt;
- die Strömungen und Perspektiven des Denkens seit der vorwissenschaftlichen Zeit bis in die Gegenwart aufzählen, sowie die jeweils wesentlichen Grundbegriffe, Problemlösungsansätze und -methoden diskutieren;
- die Notwendigkeit ethischen Handelns begründen, und können Methoden anwenden, mit denen ethische Fragestellungen systematisch behandelt werden;
- die Verantwortung der Informatik bei der Gestaltung von Technologien im gesellschaftlichen Wandel diskutieren;
- wesentliche Ereignisse und Ideen aus der Geschichte der Informationstechnologien aufzählen und deren Relevanz kritisch reflektieren.

*Kognitive und praktische Kompetenzen:* Durch die theoretische und praktische Auseinandersetzung mit den Inhalten werden folgende kognitive Fertigkeiten vermittelt:

- Auswahl und Einsatz von Strategien, Methoden und Werkzeugen zur Anwendung verschiedener Denk- und Problemlösungsformen;
- Formulierung von Kritik aus unterschiedlichen Perspektiven, rationale Auseinandersetzung im kritischen Dialog;
- Einbettung aktueller Entwicklungen und Technologien in einen historisch-kritischen Kontext
- selbständige Wissenssuche und Wissenserwerb
- Kritische Reflexion

*Soziale Kompetenzen und Selbstkompetenzen:* Gruppenarbeiten in verschiedenen Zusammensetzungen und Gruppengrößen erlauben Studierenden Erfahrungen zu sammeln, wie an Problemstellungen gemeinschaftlich herangegangen werden kann. In peer-review Aufgaben lernen Studierende, konstruktive Kritik an der Arbeit anderer zu üben, solche auch anzunehmen, und diese effektiv in ihre eigene Arbeit einfließen zu lassen. Die unterschiedlichen Herangehensweisen an Probleme eröffnen Studierenden darüber hinaus Handlungsoptionen und Sichtweisen, die einen kreativen und innovativen Zugang zur Gestaltung von Technologie erlauben. Dadurch wird auch zu ethischem Verhalten in Informatik und Gesellschaft angeregt.

### **Inhalt:**

- Vorwissenschaftliche Denkweisen
- Denkweisen der naturwissenschaftlichen Revolution
- Mathematisches Denken, insbesondere Rekursion, Abstraktion, Induktion und Deduktion
- Computational Thinking inklusive der Fragen der Berechenbarkeit
- Design Thinking, mit einem Schwerpunkt des Mottos der TU, „Technik für Menschen“

- Kreativität und Innovation
- Kritisches Denken, mit besonderer Betonung von Bias und algorithmic Bias
- Verantwortung und Ethik, Verhaltensregeln, code of conducts, Freiheit der Forschung
- Organisation und Struktur der TU Wien sowie der Fakultät für Informatik
- Bachelor- und Masterstudien der Informatik
- Forschungsgebiete der Informatik (der Fakultät und allgemein)
- Strategien für einen erfolgreichen Studienabschluss (Lernen und Lernstrategien, soziales Lernen, Stressbewältigung, Umgang mit Krisen)

In die Behandlung dieser Themen werden folgende Inhalte übergreifend behandelt:

- Geschichte der Informatik
- Informatik und Gesellschaft
- Lernen und Forschen an der TU Wien
- Informatik als Wissenschaft
- Diversität und Genderkompetenz

**Erwartete Vorkenntnisse:** Keine.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen von unterschiedlichen Vortragenden vorgestellt und teilweise von Studierenden selbst erarbeitet. In selbstorganisierter Arbeit bearbeiten die Studierenden in einem eigenen Online-System Übungsaufgaben und begutachten im double blind peer reviewing-Verfahren die Arbeit von Mitstudierenden. Zur Bewertung werden nicht nur die Leistungen in den Übungsaufgaben, sondern auch die Qualität des Reviewing herangezogen. Die Beurteilung des Orientierungsteils erfolgt auf Basis eines Online-Tests.

**Lehrveranstaltungen des Moduls:**

5,5/4,0 VU Denkweisen der Informatik

1,0/1,0 VU Orientierung Informatik und Wirtschaftsinformatik

## **Einführung in Artificial Intelligence**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden fundamentale Konzepte die zum Verständnis der Arbeitsweise als auch zur Erstellung von Systemen der Artificial Intelligence von Bedeutung sind benennen und erläutern und theoretische Zusammenhänge korrekt argumentieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- die eingesetzten Techniken und Methoden formal analysieren,
- Methoden und Techniken für eine vorgegebene Aufgabenstellung zielgerichtet auswählen, sowie
- Lösungen kritisch bewerten.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden erarbeitete Lösungen kommunizieren.

**Inhalt:**

- Einführung und Geschichte;
- Intelligente Agenten;
- Suchverfahren (uninformierte Suche, informierte Suche, lokale Suche);
- Constraint Satisfaction Probleme (CSP);
- Aspekte der Wissensrepräsentation (Schließen, Ontologisches Engineering);
- Planen;
- Maschinelles Lernen (Lernen von Beispielen, Neuronale Netzwerkarchitekturen, Deep Learning);
- Entscheidungstheoretische Konzepte (Utility-Theorie, Decision Networks und Information);
- Philosophische und Ethische Aspekte der AI.

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls

- elementare Beweistechniken, wie sie im Rahmen der einführenden Mathematikveranstaltungen vermittelt werden, sowie
- die wesentlichen Konzepte in Datenstrukturen und Algorithmen beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls

- Beweisprinzipien zur Erstellung eigener Beweise korrekt anwenden,
- die Korrektheit der einzelnen Beweisschritte formal argumentieren, sowie
- vorgegebene Problembeschreibungen algorithmisch umsetzen können.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Algorithmen und Datenstrukturen, Grundzüge digitaler Systeme, Theoretische Informatik.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Lehrveranstaltung des Moduls besteht aus einem Vorlesungsteil (Frontalvortrag) und einem begleitenden Übungsteil. Die Vorlesung dient zur Vermittlung der theoretischen

Grundlagen des besprochenen Fachgebietes während in der Übung die Teilnehmer in selbständiger Weise Lösungen zu konkreten Aufgabenstellungen erarbeiten. Die Beurteilung der Vorlesung erfolgt auf Basis von Prüfungen (schriftlich und/oder mündlich) und die Beurteilung der Übung anhand der abgegebenen Lösungen der Aufgabenstellungen sowie mittels Abgabegesprächen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Einführung in Artificial Intelligence

## **Einführung in die Programmierung**

**Regelarbeitsaufwand:** 9,5 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden Folgendes beschreiben:

- systematische Vorgehensweisen bei der Programmierung (einschließlich Erstellen, Nachvollziehen, Debuggen, Modifizieren und Dokumentieren von Programmen),
- wichtige Konzepte einer aktuellen alltagstauglichen Programmiersprache,
- ausgewählte Algorithmen, Datenstrukturen und Datenabstraktionen,
- häufige Fehlerquellen und Techniken zur Qualitätssicherung.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- Inhalte natürlichsprachiger Programmieraufgaben in ausführbare Programme umsetzen,
- Vorgehensweisen und Werkzeuge beim Programmieren systematisch anwenden,
- beschriebene Datenabstraktionen, Algorithmen und Datenstrukturen implementieren,
- einfache Maßnahmen zur Verbesserung der Qualität von Programmen anwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- Programmieraufgaben selbständig lösen sowie in Zweierteams zusammenarbeiten,
- Programmeigenschaften kommunizieren.

**Inhalt:**

- Prozedurale Programmierkonzepte (Variablen, Datentypen, Operatoren, Verzweigungen, Schleifen, Arrays, Unterprogramme)
- Fundamentale Entwicklungsmethoden (prozedurale Abstraktion, dynamisches und statisches Programmverstehen, Prüfen auf Korrektheit, Debugging) und Programmierwerkzeuge einschließlich einer Programmierumgebung

- Rekursion
- Ein- und Ausgabe mit Überprüfung von Eingaben
- Datenabstraktion
- Implementierung und wesentliche Eigenschaften rekursiver Datenstrukturen (Listen und Bäume)
- Grundlegende Algorithmen (Einfügen, Löschen, Suchen, Sortieren, Vergleichen, Konvertieren) für verschiedene Datenstrukturen
- Abstraktion über Datenstrukturen mit vergleichbaren Zugriffsfunktionen
- Exception-Handling
- Einfache Testmethoden und Code-Review
- Ansätze zur Programmoptimierung
- Programmierstile und Programmdokumentation

**Erwartete Vorkenntnisse:** Keine.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Übungsaufgaben sind vorwiegend Programmieraufgaben. Sie werden zu einem Teil örtlich ungebunden (für einige Aufgaben in Zweiertteams) innerhalb vorgegebener Fristen, zum anderen Teil unter kontrollierten Bedingungen selbständig gelöst. Die Beurteilung erfolgt auf Basis schriftlicher Tests und kontinuierlich in Übungen erbrachter Leistungen.

**Lehrveranstaltungen des Moduls:**

5,5/4,0 VU Einführung in die Programmierung 1

4,0/3,0 VU Einführung in die Programmierung 2

## **Einführung in paralleles Rechnen (Parallel Computing)**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Das Modul vermittelt Prämisse und Ziele des parallelen Rechnens und gibt Einblicke in die Leistungsbewertung und -analyse von parallelen Algorithmen und Verfahren. Vermittelt werden allgemeine algorithmische Techniken zur Parallelisierung, grundlegende Eigenschaften paralleler Rechnerarchitekturen und elementare Fähigkeiten des parallelen Programmierens anhand konkreter Programmiermodelle, Programmiersprachen und -schnittstellen.

*Fachliche und methodische Kompetenzen:* Studierende erwerben fundierte Kenntnisse des parallelen Rechnens, insbesondere der Leistungsbewertung eines parallelen Algorithmus und dessen Implementierung. Studierende erwerben Kenntnisse von Schnittstellen und Sprachen zur Implementierung von parallelen Algorithmen, sowie deren Zusammenspiel auf unterschiedlichen Parallelrechnerarchitekturen, einschließlich einiger etablierter

Schnittstellen, wie z.B. OpenMP, Cilk und MPI (das „Message-Passing Interface“). Studierende erwerben einführende Kenntnisse in grundlegende algorithmische Werkzeuge und die Grenzen der Parallelisierbarkeit werden erörtert.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können Studierende selbständig

- angeben, wie die Leistung eines parallelen Algorithmus theoretisch und praktisch zu beurteilen ist,
- anhand vorgegebener Algorithmen und Implementierungen beurteilen, inwieweit diese effizient parallelisiert worden sind oder werden können,
- anhand von Problembeschreibungen und existierenden sequentiellen Algorithmen, parallele Lösungsansätze angeben, und eventuelle Grenzen dieser Ansätze angeben,
- einfache parallele Algorithmen selber für eine dafür geeignete Schnittstelle entwerfen und hierbei unterschiedliche Parallelisierungskonzepte, Schnittstellen und Sprachen berücksichtigen,
- entwickelte Algorithmen mit Hilfe einer Schnittstelle korrekt implementieren und zur Lauffähigkeit bringen,
- mittels Experimenten und Messungen die Güte der Umsetzung beurteilen.

*Soziale Kompetenzen und Selbstkompetenzen:* Studierenden lernen, algorithmische Probleme zu formulieren und Ansätze der Parallelisierung zu entwickeln und diese selbständig (oder in kleineren Gruppen) korrekt zu beschreiben.

#### **Inhalt:**

- Asymptotische Komplexität, Speed-up, Effizienz, Amdahlsches Gesetz.
- Parallelrechnerarchitekturen mit gemeinsamen und verteilten Speicher („shared and distributed memory“), Hochleistungsrechnerarchitekturen.
- Algorithmische Muster und Probleme wie z.B. Schablone („Stencil“), Präfixsumme, Mischen, Sortieren, und allgemeine Ansätze zur Parallelisierung.
- Theoretische und experimentelle Leistungsanalyse und -beurteilung.
- Datenaustausch und Kommunikationsprobleme, kollektive Kommunikationsoperationen.
- Synchronisationsprobleme und Vermeiden von Synchronisation.
- Einfache untere und obere Schranken für wichtige Kommunikationsprobleme.
- Modelle der Parallelität wie „Threads“, Prozesse, Aufgaben („Tasks“).
- Unterscheidung zwischen Daten- und Aufgaben-Parallelität.
- Die Schnittstelle und Spracherweiterung OpenMP.
- Die Spracherweiterung Cilk.
- Die Schnittstelle MPI.
- Weitere Schnittstellen und Sprachen für das parallele Programmieren.

#### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Grundlegende Kenntnisse in Algorithmen und Datenstrukturen, Rechnerarchitekturen, Programmiersprachen und Betriebssysteme werden erwartet und zum Teil vorausgesetzt.

*Kognitive und praktische Kompetenzen:* Programmierung in C oder ähnlicher Sprache. Grundlegende Methoden der Software-Entwicklung. Einfache asymptotische Laufzeitanalyse von Algorithmen und Datenstrukturen.

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls

- Programmieraufgaben selbständig lösen und
- in Zweiertteams zusammenarbeiten können.

Dieses Modul baut auf den Kenntnissen und Fertigkeiten folgender Module auf: Algorithmen und Datenstrukturen, Betriebssysteme, Einführung in die Programmierung, Programmierparadigmen, Grundzüge digitaler Systeme

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden durch obligatorische Vorlesungen vermittelt, welche durch einen Foliensatz und begleitende Literatur unterstützt werden. Die Projekte sind kleinere Programmieraufgaben, in denen vorgegebene Probleme in mehreren der vorgestellten Schnittstellen implementiert sowie theoretisch und praktisch analysiert werden sollen. Vorgaben für die Lösungen sowie zur Art und Form der Abgabe werden gegeben. Abgabefristen werden ebenfalls vorgegeben und sind bindend. Für die experimentelle Auswertung wird der Zugriff auf Parallelrechner gewährleistet. Die Beurteilung erfolgt anhand der abgegebenen schriftliche Projektlösungen, sowie einer mündliche oder auch schriftlichen Prüfung.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Einführung in paralleles Rechnen (Parallel Computing)

## **Einführung in Security**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Nach positiver Absolvierung dieser Einführung besitzen die Studierenden Grundkenntnisse der IT-Sicherheit. Die Studierenden können Aspekte der IT-Sicherheit in Projekten identifizieren und können Maßnahmen setzen, um diese zu berücksichtigen.

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls kennen und verstehen die Studierenden

- die theoretischen Grundlagen der IT-Sicherheit,
- die theoretischen Grundlagen der Kryptographie,
- wichtige Sicherheitsaspekte in IT-Projekten,
- wichtige Best-Practice Sicherheitsmaßnahmen

und können diese erklären.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- wichtige Best-Practice Sicherheitsmaßnahmen umsetzen,
- die Denkweise von Angreifer\_innen verstehen und
- die vorgestellten Angriffstechniken entwickeln.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls besitzen die Studierenden

- Aufmerksamkeit für Sicherheitsaspekte in IT-Projekten und
- Aufmerksamkeit für beweisbare Sicherheit und können
- die Relevanz von IT-Sicherheit und von Lösungsideen für IT-Sicherheitsproblemen kommunizieren.

### **Inhalt:**

- Sicherheitsprinzipien (Vertraulichkeit, Integrität, Erreichbarkeit)
- Authentifikation und Zugangskontrollverfahren
- Systemsicherheit: Buffer- und Stack-Overflows, Gegenmaßnahmen gegen Overflow-Sicherheitslücken (N+X, Stack Canaries, ASLR), rücksprungrorientierte Programmierung, Tools und Bibliothek zum Entwickeln von binären Exploits
- Websicherheit: Einführung ins Web (Protokolle, Programmiersprachen), serverseitige Angriffe (SQL Injections, NoSQL injections, Befehlsinjektionen), clientseitige Angriffe (Cross-Site Request Forgery, Cross-Site Scripting), Gegenmaßnahmen (CSRF Tokens, Prepared Statements, usw.), im Browser implementierte Sicherheitsmechanismen (Same-Origin-Policy, Content-Security-Policy, Cookie flags), Tools und Bibliothek zum Testen und Ausnutzen von Webanwendungen (requests, Burp Suite)
- Grundlagen von Informationsfluss und Typsystemen
- Netzwerk- und Kommunikationssicherheit: Grundlagen der Kryptographie (symmetrische und asymmetrische Verfahren, Betriebsarten, Hash-Funktionen, MACs und Signaturen), kryptographische Protokolle, TLS, Modellierung und Verifikation von Protokollen

### **Erwartete Vorkenntnisse:**

Dieses Modul baut auf die Kenntnissen und Fertigkeiten folgender Module auf: Betriebssysteme, Einführung in die Programmierung, Grundzüge digitaler Systeme

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte des Kurses werden in einem Vorlesungsteil durch Frontalvortrag vermittelt und in begleitenden Übungen weiter vertieft. In den Übungen werden den Studierenden angreifbare Anwendungen zur Verfügung gestellt, die mit den im Kurs erworbenen Angriffstechniken ausgenutzt werden können. Die Beurteilung setzt sich aus der Leistung in einer Abschlussprüfung und den Berichten zu den Übungen zusammen.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Einführung in Security

# Einführung in Visual Computing

**Regelarbeitsaufwand:** 6,0 ECTS

## Lernergebnisse:

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls verstehen Studierende die grundlegende Techniken der wichtigsten Visual Computing Bereiche und haben ein kritisches Verständnis ihrer Theorien und Grundsätze erworben:

- Computergraphik,
- Computer Vision,
- Digitale Bildverarbeitung,
- Visualisierung,
- Geometrische Modellierung.

*Kognitive und praktische Kompetenzen:* Durch die praktische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie modernen Programmiersprachen und Entwicklungsumgebungen) können Studierende nach positivem Abschluss des Moduls:

- formale Grundlagen und Methoden zur Modellbildung, Lösungsfindung und Evaluation einsetzen,
- an einschlägige Probleme methodisch fundiert herangehen, insbesondere in offenen/unspezifizierten Problemsituationen,
- Standard-Entwurfs- und Implementierungsstrategien anwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Ein Schwerpunkt liegt in der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale. Studierende werden geschult in

- Eigeninitiative und Neugierde,
- Selbstorganisation, Eigenverantwortlichkeit,
- Problemformulierungs- und Problemlösungskompetenz,
- Kenntnisse der eigenen Fähigkeiten und Grenzen, Kritikfähigkeit.

## Inhalt:

- Digitale Bilder: Auflösung, Abtastung, Quantisierung, Farbrepräsentation
- Bildoperationen: Punktoperationen, lokale und globale Operationen
- Segmentierung
- Bewegungserkennung
- Repräsentation: konturbasiert, regionenbasiert (Momente, Graphen)
- Kodierung: Entropie-Kodierung, Source-Kodierung
- Komprimierung: Prediktive Kodierung, Vektorquantisierung, JPEG, MPEG
- Hardware: Ein- und Ausgabegeräte, Bildgebende Verfahren, Sensoren
- Radiometrische und Geometrische Transformationen

- Graphikprimitive und deren Attribute
- 2D- und 3D-Viewing, Graphikarchitektur (Rendering Pipeline, etc)
- Sichtbarkeitsverfahren
- 3D Objektrepräsentationen
- Kurven und Flächen
- Licht und Schattierung
- Ray-Tracing und Globale Beleuchtung
- Texturen und andere Mappings
- Farben und Farbmodelle
- Computational Photography
- Geometrische Modellierung

**Erwartete Vorkenntnisse:**

- Mathematik auf Maturaniveau (Vektorrechnung, Winkelfunktionen, Differenzieren und Integrieren, lineare Algebra, einfache Geometrie)
- Elementare Programmierkenntnisse

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Einführung in die Programmierung

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Vorlesung mit Übung: Angesichts der großen Anzahl von HörerInnen ist das eine Vorlesung mit Unterstützung durch Medien (hauptsächlich Datenprojektor), in die Übungsbeispiele eingebaut sind. Es gibt ein kompaktes Skriptum, außerdem werden Kopien der Vorlesungsfolien zur Verfügung gestellt. Die Leistungsbeurteilung erfolgt durch die erfolgreiche Abgabe von Übungsbeispielen und die erfolgreiche Teilnahme an zwei Tests.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Einführung in Visual Computing

**Einführung in wissensbasierte Systeme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden fundamentale Konzepte, die zum Verständnis der Arbeitsweise als auch zur Erstellung von wissensbasierten Systemen von Bedeutung sind, benennen und erläutern und theoretische Zusammenhänge korrekt argumentieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- die eingesetzten Techniken und Methoden formal analysieren,

- Methoden und Techniken für eine vorgegebene Aufgabenstellung zielgerichtet auswählen, sowie
- Lösungen kritisch bewerten.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden erarbeitete Lösungen verständlich kommunizieren.

**Inhalt:**

- Einführung und geschichtlicher Hintergrund;
- Prädikatenlogik als Spezifikationsprache;
- Description Logik und Ontologisches Schließen;
- Nichtmonotones Schließen;
- Answer-Set Programmierung;
- Probabilistische Verfahren;
- Entwicklung von wissensbasierten Systemen.

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls elementare Beweistechniken, wie sie im Rahmen der einführenden Mathematikveranstaltungen vermittelt werden, beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls

- elementare mathematische Beweisprinzipien zur Erstellung eigener Beweise korrekt anwenden,
- die Korrektheit der einzelnen Beweisschritte formal argumentieren, sowie
- vorgegebene Problembeschreibungen algorithmisch umsetzen können.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Algorithmen und Datenstrukturen, Grundzüge digitaler Systeme, Theoretische Informatik.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Lehrveranstaltung des Moduls besteht aus einem Vorlesungsteil (Frontalvortrag) und einem begleitenden Übungsteil. Die Vorlesung dient zur Vermittlung der theoretischen Grundlagen des besprochenen Fachgebietes während in der Übung Lösungen zu konkreten Aufgabenstellungen in selbständiger Weise erarbeitet werden müssen. Die Beurteilung der Vorlesung erfolgt auf Basis von Prüfungen (schriftlich und/oder mündlich) und die Beurteilung der Übung anhand der abgegebenen Lösungen der Aufgabenstellungen sowie mittels Abgabegesprächen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Einführung in wissensbasierte Systeme

## Freie Wahlfächer und Transferable Skills

**Regelarbeitsaufwand:** 18,0 ECTS

**Lernergebnisse:** Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

**Inhalt:** Abhängig von den gewählten Lehrveranstaltungen.

**Erwartete Vorkenntnisse:** Abhängig von den gewählten Lehrveranstaltungen.

**Verpflichtende Voraussetzungen:** Abhängig von den gewählten Lehrveranstaltungen.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Abhängig von den gewählten Lehrveranstaltungen.

**Lehrveranstaltungen des Moduls:** Die Lehrveranstaltungen dieses Moduls können frei aus dem Angebot an wissenschaftlichen und künstlerischen Lehrveranstaltungen, die der Vertiefung des Faches oder der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen dienen, aller anerkannten in- und ausländischen postsekundären Bildungseinrichtungen ausgewählt werden, mit der Einschränkung, dass zumindest 6 ECTS aus den Themenbereichen der Transferable Skills zu wählen sind. Für die Themenbereiche der Transferable Skills werden insbesondere Lehrveranstaltungen aus dem Wahlfachkatalog „Transferable Skills“ der Fakultät für Informatik (Anhang F) und aus dem zentralen Wahlfachkatalog der TU Wien für „Transferable Skills“ empfohlen.

## Funktionale Programmierung

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden die als Modulinhalt angeführten Konzepte und Techniken mit fachspezifischer Terminologie beschreiben.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- die als Modulinhalt angeführten Techniken anwenden,
- in natürlicher Sprache (unvollständig) beschriebene Programmieraufgaben in aufgabenangemessene ausführbare funktionale Programme umsetzen.
- die Bedeutung dieser Programme erklären und schrittweise auch mit Papier und Bleistift für verschiedene Auswertungsstrategien funktionaler Sprachen ausführen.
- eigene (eventuell auch fremde) Programme nach vorgegebenen Kriterien kritisch beurteilen.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden Programmieraufgaben allein und auch im Team lösen.

**Inhalt:**

- Typische Konzepte funktionaler Programmiersprachen, insbesondere Funktionen höherer Ordnung, Polymorphie auf Funktionen und Datentypen
- Sprachkonzepte applikativer und funktionaler Programmierung, fleißige und faule Auswertungsordnungen funktionaler Sprachen, potentiell unbeschränkte Datenstrukturen, Grundzüge von Typprüfung, Typinferenz, dem Lambda-Kalkül als Grundlage funktionaler Programmierung.
- Techniken zur produktiven Verwendung dieser Konzepte

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls

- systematische Vorgehensweisen beim Programmieren und
- wichtige Konzepte einer aktuellen alltagstauglichen Programmiersprache und
- Programmierparadigmen

beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls

- Inhalte natürlichsprachiger Programmieraufgaben in ausführbare Programme umsetzen,
- Vorgehensweisen und Werkzeuge beim Programmieren systematisch anwenden,
- beschriebene Datenabstraktionen, Algorithmen und Datenstrukturen implementieren und
- Basistechniken der funktionalen und objektorientierten Programmierung anwenden können.

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls

- Programmieraufgaben selbständig lösen und
- allein und in kleinen Teams zusammenarbeiten können.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Einführung in die Programmierung, Programmierparadigmen.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Übungsaufgaben sind vorwiegend Programmieraufgaben, die innerhalb vorgegebener Fristen allein und teilweise in Teams zu lösen sind. Die Beurteilung erfolgt auf Basis einer kontinuierlichen Überprüfung der Lösungen dieser Aufgaben sowie durch Prüfung(en) bzw. Test(s).

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Funktionale Programmierung

# Grundlagen der Computergraphik

**Regelarbeitsaufwand:** 6,0 ECTS

## Lernergebnisse:

*Fachliche und methodische Kompetenzen:* Nach Absolvierung dieses Moduls können Studierende:

- tiefere Inhalte im Bereich Computergraphik, aufbauend auf das Modul „Einführung in Visual Computing“, wiedergeben.

*Kognitive und praktische Kompetenzen:* Durch die Umsetzung einer umfangreichen graphischen Anwendung können Studierende nach Absolvierung des Moduls:

- eine höhere Programmiersprache einsetzen
- ein hardwarenahes Graphik-API (OpenGL oder DirectX) inklusive Shading Language (GLSL, HLSL) einsetzen
- Computergraphik in praxisnahen Anwendungsgebieten einsetzen.

*Soziale Kompetenzen und Selbstkompetenzen:* Das Modul fördert

- die Arbeit in kleinen Teams,
- die Fähigkeit, eine interaktive Software vor vielen Leuten zu präsentieren und
- die Fähigkeit, eigenständig Ziele zu definieren.

Außerdem fördert das Modul die Kreativität und graphische Ausdrucksfähigkeit.

**Inhalt:** In einer Vorlesung werden fortgeschrittene Methoden der Computergraphik vertiefend behandelt:

- Höhere graphische Programmierung (z.B. Unity)
- Höhere Modellierungstechniken
- Komplexe Datenstrukturen für graphische Daten
- Abtasten und die Rekonstruktion kontinuierlicher Signale
- Texturierung
- Digital Fabrication
- Illustrative Visualisierung
- Computational Photography

## Erwartete Vorkenntnisse:

*Fachliche und methodische Kompetenzen:*

- Mathematik: Vektor und Matrizenrechnung, aus Lineare Algebra
- Algorithmen und Datenstrukturen, aus dem gleichnamigen Modul
- Programmieren, objektorientiertes Programmieren
- Computergraphik Kenntnisse, aus Modul Einführung in Visual Computing (beide Teile) sowie dessen Voraussetzungsmodul

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:**

- 3 ECTS Vorlesung (Frontalvortrag)
- 3 ECTS Übung

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Grundlagen der Computergraphik

## **Grundzüge digitaler Systeme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- Zahlendarstellungen unterscheiden und nutzen,
- grundlegende Konzepte der Informations- und Codierungstheorie beschreiben und anwenden,
- Ausdrücke der Booleschen Algebra formulieren, manipulieren und interpretieren sowie Minimierungsverfahren darauf anwenden,
- grundlegende logische Konzepte nachvollziehen, anwenden und übertragen,
- einfache Schaltnetze entwerfen und erklären,
- ein System mit Hilfe einer geeigneten Logik oder eines geeigneten Automatentyps beschreiben,
- syntaktische und semantische Fehler in einem Modell erkennen und korrigieren,
- informell beschriebene Systeme analysieren, auf die relevanten Merkmale reduzieren und mit formalen Spezifikationsmethoden modellieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden methodische Ansätze auf konkrete Aufgabenstellungen umsetzen. Insbesondere sind sie in der Lage

- Auswirkungen verschiedener Spezifikations- und Abbildungsmethoden auf die Modellierung eines Systems zu analysieren,
- eine für die Problemstellung geeignete Abstraktionsebene sowie Spezifikationsmethode zu wählen und darauf eine passende Modellierung zu erstellen,
- eine fremde Modellierung zu analysieren und entsprechend Verbesserungsvorschläge zu formulieren,
- formal-mathematische Beschreibungen anzuwenden und in eigenen Worten zu erklären,
- die erlernten Formalismen auch im Kontext der in der Fachliteratur üblichen Varianten dieser Definitionen zu verwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Die Studierenden können umfangreichere Aufgabenstellungen eigenverantwortlich lösen und beherrschen Selbstorganisation und Zeitmanagement, um die Lösungen fristgerecht fertig zu stellen. Sie können ihre eigenen Gedanken, Lösungen und Modelle argumentieren und kommunizieren sowie mit anderen Personen professionell diskutieren.

**Inhalt:**

- Zahlendarstellung und Zahlencodierung
- Gleitkommaarithmetik
- Grundkonzepte der Informations- und Codierungstheorie
- Logik in der Informatik
- Aussagenlogik, Boolesche Algebra, Minimierungsverfahren
- Prädikatenlogik als Spezifikationsprache
- Endliche Automaten, inklusive Moore- und Mealy-Automaten
- Reguläre Ausdrücke und kontextfreie Grammatiken
- Schaltnetze
- Realisierungen von Automaten
- Petri-Netze

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls Mathematik auf AHS/BHS-Maturaniveau anwenden können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls fachliche Texte auf AHS/BHS-Maturaniveau verstehen können.

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls grundlegendes Selbstmanagement anwenden können.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Vorträge und Repetitorien; Unterstützung durch eine Online-Plattform mit betreutem Forum und Streaming; Übung in Gruppen zur Vertiefung und Festigung des Lehrstoffes. Die Beurteilung erfolgt auf Basis schriftlicher Tests und kontinuierlich in Übungen und auf der Online-Plattform erbrachter Leistungen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Grundzüge digitaler Systeme

## **Interface und Interaction Design**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Studierende verstehen die Notwendigkeit der systematischen Berücksichtigung der Interessen von Nutzerinnen und Nutzern wie auch von anderen Interessensgruppen in der Gestaltung technischer Systeme.
- Studierende können methodische Werkzeuge benennen und anwenden, die diese Berücksichtigung ermöglichen.
- Sie können den Wert von Design Guidelines und Design Patterns für die Gestaltung interaktiver Systeme einschätzen, und verstehen die Abhängigkeiten zwischen interaktiven und nicht-interaktiven Komponenten eines technischen Systems.
- Studierende können nachvollziehen, warum Anforderungen und Bedürfnisse von Nutzer\_innen oft widersprüchlich und unvollständig sind, und kennen Wege, um damit trotzdem produktiv umgehen zu können.

*Kognitive und praktische Kompetenzen:*

- Studierende verstehen das Wechselspiel zwischen Problem Setting und Problem Solving, und verstehen, wie man diese beiden Arbeiten im produktiven Wechselspiel einsetzt.
- Sie können Methoden des nutzungsorientierten Designs anwenden.
- Teilnehmer\_innen können Werkzeuge des inklusiven, nutzerorientierten und partizipativen Designs anwenden.
- In projektbasierter Arbeit kann der Wert der theoretisch erworbenen Kenntnisse erprobt und praktisch erfahren werden.

*Soziale Kompetenzen und Selbstkompetenzen:* Studierende verstehen die Komplexität der Bedürfnisse der Nutzer\_innen, für die sie Gestalten. Sie können kreative und innovative Ideen entwickeln, um so aussergewöhnliche Lösungen zu finden. Sie üben die Arbeit in Gruppen, und übernehmen Verantwortung für die Gestaltung technischer Systeme.

**Inhalt:**

- Wesentliche Aspekten der Planung, Gestaltung und Entwicklung von Benutzerschnittstellen für interaktive Systeme
- Grundlagen von User Interface Design
- Verständnis für Kriterien guter bzw. schlechter User Interfaces
- grundlegende Designkonzepte und Gestaltungsprinzipien für die erfolgreiche Umsetzung von Benutzerschnittstellen
- Methoden, Prozesse und Hilfsmittel für erfolgreiches Interface & Interaction Design
- Ausblick auf aktuelle Trends und neue Interaktionskonzepte und Interfaces

**Erwartete Vorkenntnisse:**

Es werden die Kenntnisse des Moduls Einführung in die Programmierung vorausgesetzt.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Dieses Modul wird in der Form einer Vorlesung mit Übungen gehalten, in der einerseits die

theoretischen Inhalte vermittelt werden, welche wiederum in Form von Übungen in Kleingruppen mit unterschiedlichen Übungsbeispielen praktisch erlernt werden. Das Modul umfasst einen Projektteil, der einerseits die Anwendung von Methoden und Techniken in einem konkreten Projekt ermöglicht und andererseits den Studierenden den Zugang wichtiger Systeme und Technologien, die für die Implementierung sozial eingebetteter Systeme notwendig sind, anbietet.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Interface und Interaction Design

## **Introduction to Numerics**

**Regelarbeitsaufwand:** 4,5 ECTS

**Lernergebnisse:** Vertrautheit der Studierenden mit den grundlegenden Konzepten algorithmisch-numerischer Lösungsmethoden, überlegte Auswahl und der effiziente Einsatz kommerzieller oder frei verfügbarer numerischer Software; Die Studierenden lernen zu erkennen, ob ein Programm eine angemessene Lösung geliefert hat und was zu tun ist, wenn dies nicht der Fall ist; Interpretation und Analyse numerisch erhaltener Lösungen.

**Inhalt:** Grundlegende Fehlerbegriffe: Datenfehler, Verfahrens- oder Diskretisierungsfehler, Rundungsfehler; Kondition mathematischer Probleme, numerische Lösung linearer und nichtlinearer Gleichungssysteme, polynomiale Interpolation und Approximation, numerische Integration, numerische Lösung von Differentialgleichungen, Design und Verwendung numerischer Algorithmen bzw. numerischer Software.

Die praktische Umsetzung und Vertiefung des Stoffes der Vorlesung erfolgt in den Übungen durch (realitätsnahe) numerische Übungsbeispiele. Diese beinhalten sowohl theoretische Aufgabenstellungen, etwa was das Design oder die Analyse numerisch stabiler Algorithmen betrifft, als auch die praktische Implementierung und das Testen und Bewerten am Computer. Standardsoftware kommt zum Einsatz (z.B. MATLAB).

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Mathematische Grundkenntnisse

*Kognitive und praktische Kompetenzen:* Programmierung mit MATLAB

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Analysis, Einführung in die Programmierung, Grundzüge digitaler Systeme

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Vermittlung der theoretischen Grundlagen erfolgt in der Vorlesung, die Erarbeitung der praktische Fertigkeiten erfolgt in den wöchentlichen Übungen.

Die Prüfung ist mündlich und beinhaltet eher theoretisch gehaltene Fragen zum Vorlesungsstoff, teilweise auch kurz gehaltene praktische Beispiele; die Beurteilung der Übungsleistung erfolgt aufgrund der Anzahl der gekreuzten Beispiele, der Tafelleistungen und der schriftlichen Ausarbeitung von Beispielen.

**Lehrveranstaltungen des Moduls:**  
4,5/3,0 VU Introduction to Numerics

## **Logic and Reasoning in Computer Science**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Dieses Modul befasst sich mit den logischen Grundlagen der Modellierung und Argumentation in der Informatik.

*Fachliche und methodische Kompetenzen:* Fundamentale Konzepte und Resultate der Computational Logic und computergestützte Beweisführung.

*Kognitive und praktische Kompetenzen:* Die Studierenden erwerben die Fähigkeit, die Struktur von logischen Beweisen und computergestützten Argumentationen zu verstehen und selbst solche zu führen und entwerfen.

*Soziale Kompetenzen und Selbstkompetenzen:* Selbständiges Lösen von Problemen.

### **Inhalt:**

- Formalisierung mit Logik
- Aussagenlogik
- Beweissysteme für die Aussagenlogik
- Das Erfüllbarkeitsproblem der Aussagenlogik (SAT-Solving)
- SAT-Solving und das Davis-Putnam-Logemann-Loveland-Verfahren (DPLL)
- Das Resolutionsverfahren in der Aussagenlogik
- Binäre Entscheidungsdiagramme (BDDs)
- Randomisierte Algorithmen und SAT Solving
- Das Erfüllbarkeitsproblem für quantifizierte Boolesche Formeln (QBF)
- Prädikatenlogik erster Stufe und Theorien erster Stufe
- Prädikatenlogik für endliche Gegenstandsbereiche
- Das Nelson-Oppen-Verfahren und Satisfiability Modulo Theory (SMT) Solving

**Erwartete Vorkenntnisse:** Kenntnisse aus “Theoretische Informatik” und “Logik und Grundlagen der Mathematik”

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Grundzüge digitaler Systeme, Theoretische Informatik.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden in einem Vorlesungsteil vorgestellt und in begleitenden Übungen von den Studierenden erarbeitet. Die Übungsaufgaben können zeitlich und örtlich weitgehend ungebunden einzeln oder in Gruppen gelöst werden. Die Lösungen werden bei regelmäßigen Treffen mit Lehrenden und TutorInnen besprochen und korrigiert. Die Beurteilung erfolgt auf Basis schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen. Der Übungsbetrieb und die Tests können computerunterstützt durchgeführt werden.

## **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Logic and Reasoning in Computer Science

## **Logik für Wissensrepräsentation**

**Regelarbeitsaufwand:** 6,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden unterschiedliche Logiken bzw. logikbasierte Formalismen zur Wissensrepräsentation benennen und erläutern, sowie theoretische Zusammenhänge korrekt argumentieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- die eingesetzten Techniken und Methoden formal analysieren,
- Methoden und Techniken für eine vorgegebene Aufgabenstellung zielgerichtet auswählen, sowie
- Lösungen und Formalismen kritisch bewerten.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden erarbeitete Lösungen kommunizieren.

### **Inhalt:**

- Klassische Logik zur Wissensrepräsentation (inkl. Probleme, Limitierungen und ontologische Aspekte)
- Formalismen zum Nichtmonotonen Schließen
- Parakonsistente Logiken
- Wissensrevision (Belief Revision)
- Modallogik

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls die wesentlichen Konzepte der Aussagen- und Prädikatenlogik erster Stufe benennen und beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass die Studierenden vor der Absolvierung des Moduls Deduktionskonzepte und Beweisprinzipien zur Erstellung eigener Beweise korrekt anwenden und die Korrektheit der einzelnen Beweisschritte formal argumentieren können.

*Soziale Kompetenzen und Selbstkompetenzen:* Keine speziellen Voraussetzungen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Einführung in Artificial Intelligence, Grundzüge digitaler Systeme, Theoretische Informatik

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:**

- Präsentation der Lehrinhalte in einem Vorlesungsteil (Frontalvortrag)
- Selbständiges Ausarbeiten von Aufgabenstellungen durch Studierende
- Präsentation der Lösungen (inkl. der benötigten Theorie)

Leistungsbeurteilung:

- Mündliche Prüfung des Vorlesungsteil,
- Bewertung der Ausarbeitungen (inkl. der Präsentation)

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Logik für Wissensrepräsentation

## **Logikprogrammierung und Constraints**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach Absolvierung dieses Moduls können Studierende die folgenden Bereiche anwenden:

- Monotoner Teil der Programmiersprache Prolog
- Constraintsprachen, insbesondere CLP(FD)
- Spezifikationsorientierte Programmierung
- Deklarative Diagnose

*Kognitive und praktische Kompetenzen:* Durch das praktische Arbeiten mit einer logikorientierten Programmiersprache beherrschen Absolventen die folgenden Fertigkeiten:

- Deklaratives Modellieren, relationale Sichtweise
- Praktische Programmierkenntnisse in einer logikorientierten Programmiersprache
- Anwendung deklarativer Lesarten zur Fehlersuche
- Verbindung und Abwägung von deklarativen und prozeduralen Sichtweisen

*Soziale Kompetenzen und Selbstkompetenzen:*

- Der Übungsbetrieb fördert das selbständige Arbeiten in Eigenverantwortlichkeit
- Mittels des logikorientierten Programmierparadigmas wird eine neue Sichtweise des Programmierens ermöglicht
- Zusammenarbeit, insbesondere bei Anwendung der Lesarten

**Inhalt:**

- Deklarative Programmierparadigmen

- Deklarative Lesarten
- Deklarative Diagnose
- Prozedurale Lesarten
- Termination
- Grammatiken
- Constraints
- Programmieren höherer Ordnung
- Lambda-Ausdrücke
- Pure I/O

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Der Vorlesungsteil wird parallel zum Übungsteil abgehalten, sodass auch die konkrete Vorgangsweise beim Programmieren und der deklarativen Fehlersuche behandelt werden kann. Der Übungsteil besteht aus vielen kleinen Beispielen. Die Leistungsbeurteilung besteht aus einer prüfungsimmanenten Beurteilung der Programmiertätigkeit und einem mündlichen Abgabegespräch. Inhalt des Abgabegesprächs sind die Lesarten von Logikprogrammen anhand konkreter Beispiele sowie deren Anwendung zur Fehlersuche.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Logikprogrammierung und Constraints

## **Multivariate und computerintensive statistische Methoden**

**Regelarbeitsaufwand:** 9,0 ECTS

**Lernergebnisse:** Vermittlung von datenorientierten, computerintensiven Methoden zur Verarbeitung komplexer Daten

*Fachliche und methodische Kompetenzen:*

- Multivariate Methoden
- Grundlagen der statistischen Simulation

*Kognitive und praktische Kompetenzen:*

- Anwendung multivariater Methoden auf konkrete Problemstellungen
- Anwendung von computerintensiven Methoden und Software auf komplexe Problemstellungen

*Soziale Kompetenzen und Selbstkompetenzen:*

- Datenorientierte Lösung von statistischen Problemstellungen
- Lösungen von Problemen mit open-source Software

**Inhalt:** Clusteranalyse, Hauptkomponenten- und Faktorenanalyse, Diskriminanzanalyse, Zufallszahlengeneratoren und Reproduzierbarkeit, MCMC (Markov Chain Monte Carlo) Methoden, Resamplingverfahren (Bootstrap, Jackknife, Kreuzvalidierung), Testen mittels statistischer Simulation, Anwendungen von Resamplingverfahren in Zeitreihen, Datenimputation und Regression.

**Erwartete Vorkenntnisse:** Grundlegende Kenntnisse der computerorientierten Statistik.

Diese Voraussetzungen werden im Modul Statistik und Wahrscheinlichkeitstheorie vermittelt.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** In der Lehrveranstaltung Multivariate Statistik werden die gängigen multivariaten Methoden formal vermittelt, und mit der Statistiksoftware R an konkreten Daten angewandt sowie Ergebnisse diskutiert.

In der Lehrveranstaltung Statistische Simulation und computerintensive Methoden (VU) werden simulationsbasierte Lösungsstrategien für komplexe Problemstellungen gelehrt. Die Verfahren der statistischen Simulation und computerintensiver Methoden werden theoretisch als auch praktisch mittels moderner freier open-source Statistiksoftware (R) vermittelt.

**Lehrveranstaltungen des Moduls:**

4,5/3,0 VO Multivariate Statistik

1,5/1,0 UE Multivariate Statistik

3,0/2,0 VU Statistische Simulation und computerintensive Methoden

## Programmierparadigmen

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls sind Studierende in der Lage

- die wichtigsten Ziele und einige typische Anwendungsbereiche und Techniken in der objektorientierten, funktionalen, nebenläufigen (concurrent) und parallelen Programmierung (Paradigmen) sowie der Modularisierung, Parametrisierung, Ersetzbarkeit und Typisierung (Konzepte) unter Verwendung fachspezifischer Terminologie zu beschreiben,
- diese Paradigmen und Konzepte und einige ihrer Ausprägungen durch ihre wesentlichen Eigenschaften klar voneinander zu unterscheiden.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls sind Studierende in der Lage

- ausgewählte, für diese Paradigmen typische Vorgehensweisen und Techniken sowie die genannten Konzepte der Programmierung in kleinen Teams in einer alltagstauglichen Programmiersprache (Java) praktisch anzuwenden,
- in natürlicher Sprache (unvollständig) beschriebene Programmieraufgaben in ausführbare Programme in einer alltagstauglichen Programmiersprache umzusetzen, die typische Merkmale vorgegebener Programmierstile aufweisen,
- eigene Programme nach vorgegebenen Kriterien kritisch zu beurteilen.

### **Inhalt:**

- Überblick über Paradigmen in der Programmierung, ihre Zielsetzungen und Anwendungsbereiche
- typische Konzepte, Techniken und Denkweisen in der objektorientierten, funktionalen, nebenläufigen und parallelen Programmierung
- Arten der Programmfaktorisierung hinsichtlich Abhängigkeiten in Daten und Prozeduren (opaque Objekte, transparente Parameter und Funktionen, unabhängige Datenblöcke)
- Arten von Modularisierungseinheiten (Module, Komponenten, Klassen, Objekte), Parametrisierung (Generizität, Annotationen, Aspekte), Ersetzbarkeit (strukturell, abstrakt, basierend auf Design-by-Contract), Typisierung (statisch vs. stark vs. dynamisch, explizit vs. implizit)
- Umgang mit kovarianten Problemen, Überladen, Multimethoden, Exceptions
- referentielle Transparenz, funktionale Formen, applikative Programmierung
- diverse Arten der Iteration über Datenmengen
- Threads, Race-Conditions, Synchronisationsmethoden, Liveness-Properties, Map-Reduce/Collect
- Zusammenarbeit mehrerer Prozesse über gemeinsame Dateien und Streams, einfache Form der Prozesssynchronisation
- Unterscheidung zwischen nebenläufiger, paralleler und verteilter Programmierung
- Beispiele in einer alltagstauglichen Programmiersprache (Java), die Stärken und Grenzen der Programmierparadigmen aufzeigen
- ausgewählte Entwurfsmuster

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls systematische Vorgehensweisen beim Programmieren und wichtige Konzepte einer aktuellen alltagstauglichen Programmiersprache (vorzugsweise Java) beschreiben können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls, in der Lage sind,

- natürlichsprachige Programmieraufgaben in ausführbare Programme umzusetzen,
- Vorgehensweisen und Werkzeuge beim Programmieren systematisch anzuwenden,
- vorgegebene Abstraktionen, Algorithmen und Datenstrukturen zu implementieren,

- Techniken der objektorientierten Modellierung anzuwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls in der Lage sind, Programmieraufgaben selbständig zu lösen und in Zweiertteams zusammenzuarbeiten.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen und Skripten vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Übungsaufgaben sind vorwiegend Programmieraufgaben, die innerhalb vorgegebener Fristen in Teams zu lösen sind. Die Beurteilung erfolgt auf Basis einer kontinuierlichen Überprüfung der Lösungen dieser Aufgaben sowie durch Prüfung(en) bzw. Test(s).

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Programmierparadigmen

## **Programm- und Systemverifikation**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Kenntnis unterschiedlicher Spezifikationsformalismen, ihrer Semantik und ihrer Anwendungsgebiete
- Kenntnis unterschiedlicher Verifikationstools
- Verständnis grundlegender Methoden der Modellierung in Hinsicht auf Verifikationsfragen
- Beispielhafte Kenntnisse zu Zertifizierung und Industriestandards in Hinsicht auf Verifikation

*Kognitive und praktische Kompetenzen:*

- Praktischer Umgang mit Spezifikationsformalismen hinsichtlich ihrer Semantik und hinsichtlich Requirement Engineering
- Praktischer Umgang mit Verifikationstools
- Praktische Modellierung und Verifikation von Systemen und Interpretation der Ergebnisse

*Soziale Kompetenzen und Selbstkompetenzen:*

- Verständnis für das Gefahrenpotential fehlerhafter Software und Hardware
- Verständnis für die Bedeutung formaler Methoden in der Produktentwicklung
- Anwendung theoretischer Konzepte auf angewandte Fragestellungen

**Inhalt:**

- Methoden der Modellierung und Spezifikation durch Logik, Automaten, Assertions, Coverage Kriterien
- Verifikationswerkzeuge, insbesondere Model Checker, Statische Analyse, Theorembeweisen, Testen
- Praktischer Umgang mit Verifikationswerkzeugen
- Grundlagen zur Zertifizierung und zu Standards in der industriellen Validierung

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Einführung in die Programmierung, Modellierung, theoretische Informatik und Mathematik.

*Kognitive und praktische Kompetenzen:* Geübter, fachgerechter Umgang mit Computerprogrammen und Konzepten der theoretischen Informatik und Mathematik.

*Soziale Kompetenzen und Selbstkompetenzen:* Fähigkeit zur selbständigen Einarbeitung in Tools anhand schriftlicher Unterlagen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Einführung in die Programmierung, Theoretische Informatik

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die in der Vorlesung vermittelten Grundlagen und Methoden werden in praktischen Übungen am Computer und auf Papier vertieft und angewandt. Die Leistungsfeststellung erfolgt durch Beispiel-Abgaben und schriftliche Tests/Prüfungen.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Programm- und Systemverifikation

## **Daten- und Informatikrecht**

**Regelarbeitsaufwand:** mindestens 6,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden die Grundstrukturen nationalen Rechts, des EU-Rechts und des Völkerrechts sowie einschlägige Zusammenhänge beschreiben und wichtige Teilgebiete des Informatikrechts bzw. aktuelle Aspekte der rechtlichen Problematik des Internet erklären. Sie sind in der Lage, die juristischen Interpretationsmethoden zu benennen und die Vorgangsweise bei der Subsumtion von Sachverhalten unter rechtliche Tatbestände darzustellen.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden in wichtigen informatikrechtlichen Zusammenhängen argumentieren und einfache Sachverhalte aus rechtlicher Perspektive analysieren. Sie sind weiters in der Lage, solche Sachverhalte unter einschlägige Tatbestände zu subsumieren und die maßgeblichen Rechtsfolgen abzuleiten.

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls sind die Studierenden in der Lage, mit Fachjurist/inn/en über zentrale rechtliche Aspekte IT-bezogener Sachverhalte kompetent zu diskutieren.

**Inhalt:**

- Grundlagen zu Staat und nationalem Recht
- Grundlagen des internationalen Rechts und des EU-Rechts
- Problematik der Regulierung von Technik (insb. IT) durch Recht
- Grundrechte in der Informationsgesellschaft
- Datenschutzgrundgesetz
- Softwarelizenzen, im Speziellen OSS-Lizenzen
- ausgewählte Probleme des materiellen Internetrechts

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorlesungseinheiten präsentiert und in begleitenden Übungseinheiten von den Studierenden angewendet sowie vertiefend erarbeitet. Die Beurteilung setzt sich zusammen aus den bei schriftlichen Klausuren einerseits und Übungsabgaben andererseits erbrachten Leistungen.

Es werden verschiedene Techniken des Blended Learning eingesetzt:

- Unterstützung durch ein e-learning-System (TUWEL)
- Frontalunterricht für die theoretischen Grundlagen
- Aufzeichnung/Podcasts von Vorträgen (technische Vorträge, CERIAS-Seminare)
- Praktische Arbeiten am Computer
- Unterstützung durch Tutoren

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Daten- und Informatikrecht

3,0/3,0 VU Vertrags- und Haftungsrecht

## **Security**

**Regelarbeitsaufwand:** mindestens 6,0 ECTS

**Lernergebnisse:** Durch die Absolvierung des Moduls erhalten die Studierenden Grundkenntnisse der IT-Sicherheit. Die Studierenden lernen Aspekte der IT-Sicherheit in Projekten zu identifizieren und Maßnahmen zu setzen, um diese zu berücksichtigen.

*Fachliche und methodische Kompetenzen:* Das Modul vermittelt

- die theoretische Grundlagen der IT-Sicherheit,
- die theoretische Grundlagen der Kryptographie,
- wichtige Sicherheitsaspekte in IT-Projekten, und
- Wissen über wichtige Best-Practice Sicherheitsmaßnahmen.

*Kognitive und praktische Kompetenzen:*

- Umsetzung von wichtigen Best-Practice Sicherheitsmaßnahmen
- Verstehen der Denkweise von AngreiferInnen
- Entwicklung kryptographischer Verfahren und Sicherheitsbeweisen

*Soziale Kompetenzen und Selbstkompetenzen:*

- Aufmerksamkeit für Sicherheitsaspekte in IT-Projekten
- Aufmerksamkeit für beweisbare Sicherheit
- Kommunikation der Relevanz von IT-Sicherheit und von Lösungsideen für IT-Sicherheitsprobleme

**Inhalt:**

- Operating systems
- Access control
- Network security
- Application security
- Database security
- Web (application) security
- Modern Cryptography
- Security definitions
- Security proofs
- Privacy
- Security and usability
- Operations security
- Physical security
- Security architecture
- Forensics
- Exploiting vulnerabilities
- Malware
- System security
- Standards, policies, best practices
- Ethics, Compliance, Legal investigations
- Risk management
- Information security
- Business continuity, disaster recovery

**Erwartete Vorkenntnisse:**

Dieses Modul baut auf den Kenntnissen und Fertigkeiten folgender Module auf: Einführung in die Programmierung, Grundzüge digitaler Systeme

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Introduction to Cryptography

6,0/4,0 VU Privacy Enhancing Technologies  
3,0/2,0 VU Security for Systems Engineering  
3,0/2,0 VU Mobile Security

## **Semistrukturierte Daten**

**Regelarbeitsaufwand:** 6,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Das Modul vermittelt breites und integriertes Wissen und Verständnis der wissenschaftlichen Grundlagen von schwach strukturierten Daten, welches wesentlich über das auf der Ebene der Universitätszugangsberechtigung vorhandene Wissen hinausgeht.
- Die Studierenden sind befähigt Problem domänen zu beherrschen, die sich durch wenig-strukturierte Information auszeichnen.
- Die Studierenden verfügen über ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Konzepte zur Entwicklung von Anwendungen, die schwach strukturierte Daten verwenden.
- Das Wissen und Verständnis der Studierenden entspricht dem Stand der Fachliteratur in diesem Bereich.

*Kognitive und praktische Kompetenzen:*

- Die Studierenden können ihr Wissen und Verstehen praktisch zur Entwicklung von Anwendungen, die schwach strukturierte Daten verwenden, einsetzen.
- Die Studierenden können für die jeweilig vorliegende Aufgabenstellung relevante Informationen sammeln, strukturieren, bewerten und interpretieren.

*Soziale Kompetenzen und Selbstkompetenzen:*

- Die Studierenden sind in der Lage ihr Wissen selbständig zu vertiefen.
- Die Studierenden können entwicklungsbezogene Positionen und Problemlösungen formulieren, sich mit InformatikerInnen und Domänenexpert\_innen darüber austauschen und Verantwortung in einem Team übernehmen.
- Die Studierenden lernen ihre eigenen Fähigkeiten und Grenzen einzuschätzen und erwerben die Kritikfähigkeit an der eigenen Arbeit.
- Die Studierenden erlernen Selbstorganisation und Eigenverantwortlichkeit zum eigenständigen Lösen von Aufgaben.

### **Inhalt:**

- Grundlagen: Begriffsdefinitionen; Grundlagen semi- und schwach-strukturierter Daten.

- Sprachen und Technologien: Grundlagen von Markup-Sprachen (z.B. XML); Grundlagen von Graph-basierte Daten (z.B. RDF); Schemasprachen; Abfragesprachen.
- Umsetzung und praktische Realisierung: Darstellung, erstellen, abfragen und transformieren von schwach strukturierten Daten.

**Erwartete Vorkenntnisse:**

Es werden die Fertigkeiten und Kenntnisse folgender Module benötigt: Einführung in die Programmierung.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden durch Vortrageeinheiten vermittelt und durch theoretische und praktische Übungen geübt. Die Beurteilung basiert auf Test über den vorgetragenen Stoff sowie auf den erbrachten Leistungen in den Übungen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Semistrukturierte Daten

## Software Engineering

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden):

- moderne agile Vorgehensmodelle in der Softwareentwicklung verstehen
- Softwareprojekte über deren gesamten Lebenszyklus verstehen
- Grundlagen der Qualitätssicherung und des Qualitätsmanagements im Kontext der Softwareentwicklung verstehen
- den Faktor Mensch in der Softwareentwicklung verstehen

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Anwendungsszenarien modellieren, etwa in der Unified Modeling Language
- Anwendungsszenarien testgetrieben entwerfen, implementieren und validieren
- mit einer modernen Softwareentwicklungsumgebung arbeiten
- ein verteiltes Sourcecodemanagement System anwenden
- ein Projekt planen und den Projektfortschritt beurteilen

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Software Engineering Probleme selbstorganisiert lösen

- Wissen aus Software Engineering Dokumentation anwenden und ergänzen

### **Inhalt:**

- Grundlagen des Software Engineerings
- Vorgehensmodelle, Prozesse und Rollen im Software Engineering
- Methoden der Softwaretechnik (aus dem IEEE Software Engineering Body of Knowledge)
- Konzepte und Methoden in Analyse der Anforderungen, Entwurf, Implementierung, Integration, Testen und Inbetriebnahme anwenden
- Grundlagen zu Modellierung von Anwendungsszenarien, etwa mittels Unified Modeling Language (UML)
- Grundlagen der Qualitätssicherung und des Qualitätsmanagements im Kontext der Softwareentwicklung
- Technische Grundlagen: Techniken und Werkzeuge
- Grundlagen des Projektmanagements (Projektplanung, Aufgabenverteilung im Team)

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:*

- Grundkenntnisse aus Algorithmen und Datenstrukturen
- Grundkenntnisse zu Datenbanksystemen

*Kognitive und praktische Kompetenzen:*

- Eine praxisrelevante Programmiersprache und -werkzeuge (z.B. Java oder C++) anwenden
- Eine integrierte Entwicklungsumgebung und Quellcodeverwaltung anwenden

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Einführung in die Programmierung

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte werden in Vorträgen vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Die Beurteilung erfolgt auf Basis schriftlicher Tests und kontinuierlich in Übungen erbrachter Leistungen.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Software Engineering

# Software Engineering Projekt

**Regelarbeitsaufwand:** 6,0 ECTS

## **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- ein modernes agiles Vorgehensmodell praxisnah verstehen
- eine Methode für verteiltes Sourcecodemanagement verstehen
- moderne Softwareentwicklungsumgebungen und Werkzeugen verstehen
- Risiken in Softwareprojekten verstehen

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- ein modernes agiles Vorgehensmodell praxisnah anwenden
- Techniken für Abstraktion und Modellbildung in der Softwaretechnik anwenden
- ein verteiltes Sourcecodemanagementsystem anwenden
- Frameworks zur Entwicklung von modernen Software-Applikationen anwenden
- Qualitätssicherung und Qualitätsmanagement im Kontext der Softwareentwicklung anwenden
- Konzepte, Modelle und Werkzeuge im Rahmen eines mittelgroßen Software-Entwicklungsprojekts auswählen und anwenden
- Software Engineering best practices in einem Projektkontext einsetzen
- ein Projekt planen und den Projektfortschritt beurteilen
- Projektergebnisse zielgruppengerecht präsentieren

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Software Engineering Probleme in einem verteilt arbeitenden Team lösen
- mit gruppendynamischen und sozialen Aspekten erfolgreich umgehen
- Entscheidungen einer Rolle in einem Projekt verantworten
- Wissen in einem mittelgroßen Team sammeln und pflegen
- Projektergebnisse präsentieren

## **Inhalt:**

- Anwenden eines praxisrelevanten Vorgehensmodells in der Softwareentwicklung
- Auswählen und Anwenden von Konzepten, Modellen und Werkzeugen im Rahmen eines mittelgroßen Softwareentwicklungsprojekts
- Anwenden von Techniken für Abstraktion und Modellbildung in der Softwaretechnik
- Herstellen und Anpassen von hochwertiger Planung und Dokumentation
- Lösen von Software Engineering Problemen in einem verteilt arbeitenden Team

- Verantworten von Entscheidungen einer fachlichen Rolle in einem Projekt
- Sammeln und Pflegen von Wissen in einem mittelgroßen Team
- Präsentieren von Projektergebnissen

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:*

- Grundkenntnisse aus Algorithmen und Datenstrukturen
- Grundkenntnisse zu Datenbanksystemen
- Objektorientierte Analyse, Design und Programmierung
- Grundlagen der Unified Modeling Language (UML)

*Kognitive und praktische Kompetenzen:*

- Eine praxisrelevante Programmiersprache und -werkzeuge (z.B. Java oder C++) anwenden
- Eine integrierte Entwicklungsumgebung und Quellcodeverwaltung anwenden
- Ein gängiges Versionskontrollsystem (z.B. Git) zur Verwaltung von Code Kontributionen im Team einsetzen.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Einführung in die Programmierung, Software Engineering

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Im Zuge der LVA wird ein Softwareprojekt in einem Team von 4 bis 6 Personen umgesetzt. Die Projektabschnitte sind innerhalb vereinbarter Fristen zu lösen. Dies wird durch TutorInnen in wöchentlichen Treffen intensiv betreut. Die Beurteilung erfolgt auf Basis einer kontinuierlichen Überprüfung der Lösungen im Zuge von Management Reviews.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 PR Software Engineering Projekt

## **Softwareprojekt-Beobachtung und -Controlling**

**Regelarbeitsaufwand:** 6,0 ECTS

### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Dieses Modul befasst sich mit der strategischen qualitätserzeugenden Supervision von SW- und IT-Projekten in den vielfältigen Anwendungsbereichen der Informatik (z.B. Betriebliche Informationssysteme, Internet-Plattformen, Versicherungs-IT, Medizinische Informatik). Einige typische methodische Schwachstellen in der Praxis sollen anhand von konkreten Projekt- und System-Beobachtungen identifiziert und erlebt werden. Ihre Behebung im konkreten Umfeld soll konzipiert und diskutiert werden. Ziel ist, den Studierenden die Ursachen des Abstandes

zwischen SE-Theorie und konkreter Praxis verständlich zu machen und in Form konkreter Fallbeispiele die Behebung erfahrbar zu machen.

*Kognitive und praktische Kompetenzen:* Das Modul vermittelt Kenntnisse über Strategien und Konzepte zur Begleitung, Bewertung und Steuerung von Software-Projekten. Es vermittelt Kenntnisse über Kriterien für erfolgreiche Projekte sowie deren Evaluierung und Beurteilung anhand etablierter Software Engineering- und Projekt-Management-Methoden.

*Soziale Kompetenzen und Selbstkompetenzen:* Das Modul behandelt die Notwendigkeit effektiver Kommunikation zur Bewertung und Steuerung von Softwareentwicklungsprojekten. Es wird die Bedeutung von pragmatischem Risikomanagement, realitätsbezogenen Projektkennzahlen und interdisziplinärem Denken für den Projekterfolg vermittelt und auf die Möglichkeiten und Herausforderungen bei der Einbindung agiler Software-Entwicklungsmethoden wie z.B. Scrum eingegangen.

#### **Inhalt:**

- Basisinhalte aus Qualitätssicherung (QS) und Usability
- Fallbeispiele aus konkreten Softwareprojekten, Schwerpunkt auf Erhebungsmethoden aus QS und Usability

#### **Erwartete Vorkenntnisse:**

Diese Modul baut auf den Kenntnissen, Fertigkeiten und Kompetenzen des Moduls Software-Qualitätssicherung auf.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** In Form von Vorlesungen mit Übungen werden die vertiefenden Inhalte vermittelt.

#### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Softwareprojekt-Beobachtung und -Controlling

## **Software-Qualitätssicherung**

**Regelarbeitsaufwand:** 6,0 ECTS

#### **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden Folgendes erklären:

- Motivation und Ziele der Softwarequalitätssicherung
- Definition und Messung von Qualität
- Organisatorische Qualitätssicherung
- Statische und dynamische Methoden der Qualitätssicherung
- Methoden zur Sicherstellung und Verbesserung der Qualität von Produkten.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Passende QS-Methoden in einem Entwicklungsprojekt auswählen
- Statische und dynamische Methoden zur Software-Qualitätssicherung anwenden

*Soziale Kompetenzen und Selbstkompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Einen Qualitätsplan präsentieren

**Inhalt:** Inhalte des Vorlesungsteils:

- Grundlagen der Software-Qualitätssicherung
- Qualitätskontrolle und Fehlerreduktion: Reviews und Inspektionen
- Dynamische Qualitätssicherung
- Organisatorische Qualitätssicherung
- Qualitätssicherungs-Standards
- Testprozess

Inhalte des Übungsteils:

- Review von Software-Modellen
- Kollaborative Code-Inspektionen
- Statische Code Analyse / Antipattern Analyse
- Test-Driven Development
- Testplanerstellung inkl. Methoden für das Ableiten von Testfällen
- Testautomatisierung Blackbox/Whitebox
- Testen in agilen Prozessen

**Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls kennen:

- Grundlagen der Mathematik und Statistik
- Grundlagen der Unified Modeling Language (UML)
- Objektorientierte Analyse, Design und Programmierung
- Grundlegende Design-Patterns in der Programmierung
- Grundkenntnisse zu Datenbanksystemen

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls können:

- Beherrschung einer praxisrelevanten Programmiersprache und -werkzeuge (z.B. Java oder C++)
- Umgang mit einer Integrierten Entwicklungsumgebung, Build Management und Quellcodeverwaltung

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls können:

- Programmieraufgaben selbständig lösen

Diese Vorkenntnisse werden in folgenden Modulen vermittelt: Einführung in die Programmierung, Programmierparadigmen, Software Engineering, Software Engineering Projekt

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Inhalte (theoretische Konzepte und methodische Grundlagen) werden in Vorträgen vorgestellt und in begleitenden Übungen von Studierenden erarbeitet. Übungsaufgaben sind vorwiegend praktische Beispiele aus den Bereichen Reviews und Testen am Computer, die innerhalb vorgegebener Fristen individuell zu lösen sind. Intensiver Einsatz von entsprechenden Werkzeugen z.B. Testautomatisierung zur Umsetzung der QS-Konzepte und -Methoden. In Workshops mit Gruppenarbeiten werden organisatorische Themen wie agile Organisation von Software-Teams erarbeitet. Die Beurteilung erfolgt auf Basis einer kontinuierlichen Überprüfung der Lösungen dieser Aufgaben, durch aktive Teilnahme an den Workshops sowie durch Prüfung(en) bzw. Test(s).

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Software-Qualitätssicherung

## **Grundzüge digitaler Systeme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden

- Zahlendarstellungen unterscheiden und nutzen,
- grundlegende Konzepte der Informations- und Codierungstheorie beschreiben und anwenden,
- Ausdrücke der Booleschen Algebra formulieren, manipulieren und interpretieren sowie Minimierungsverfahren darauf anwenden,
- grundlegende logische Konzepte nachvollziehen, anwenden und übertragen,
- einfache Schaltnetze entwerfen und erklären,
- ein System mit Hilfe einer geeigneten Logik oder eines geeigneten Automatentyps beschreiben,
- syntaktische und semantische Fehler in einem Modell erkennen und korrigieren,
- informell beschriebene Systeme analysieren, auf die relevanten Merkmale reduzieren und mit formalen Spezifikationsmethoden modellieren.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden methodische Ansätze auf konkrete Aufgabenstellungen umsetzen. Insbesondere sind sie in der Lage

- Auswirkungen verschiedener Spezifikations- und Abbildungsmethoden auf die Modellierung eines Systems zu analysieren,
- eine für die Problemstellung geeignete Abstraktionsebene sowie Spezifikationsmethode zu wählen und darauf eine passende Modellierung zu erstellen,
- eine fremde Modellierung zu analysieren und entsprechend Verbesserungsvorschläge zu formulieren,
- formal-mathematische Beschreibungen anzuwenden und in eigenen Worten zu erklären,
- die erlernten Formalismen auch im Kontext der in der Fachliteratur üblichen Varianten dieser Definitionen zu verwenden.

*Soziale Kompetenzen und Selbstkompetenzen:* Die Studierenden können umfangreichere Aufgabenstellungen eigenverantwortlich lösen und beherrschen Selbstorganisation und Zeitmanagement, um die Lösungen fristgerecht fertig zu stellen. Sie können ihre eigenen Gedanken, Lösungen und Modelle argumentieren und kommunizieren sowie mit anderen Personen professionell diskutieren.

#### **Inhalt:**

- Zahlendarstellung und Zahlencodierung
- Gleitkommaarithmetik
- Grundkonzepte der Informations- und Codierungstheorie
- Logik in der Informatik
- Aussagenlogik, Boolesche Algebra, Minimierungsverfahren
- Prädikatenlogik als Spezifikationsprache
- Endliche Automaten, inklusive Moore- und Mealy-Automaten
- Reguläre Ausdrücke und kontextfreie Grammatiken
- Schaltnetze
- Realisierungen von Automaten
- Petri-Netze

#### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls Mathematik auf AHS/BHS-Maturaniveau anwenden können.

*Kognitive und praktische Kompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls fachliche Texte auf AHS/BHS-Maturaniveau verstehen können.

*Soziale Kompetenzen und Selbstkompetenzen:* Es wird erwartet, dass Studierende vor Absolvierung des Moduls grundlegendes Selbstmanagement anwenden können.

**Verpflichtende Voraussetzungen:** Keine.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Vorträge und Repetitorien; Unterstützung durch eine Online-Plattform mit betreutem Forum und Streaming; Übung in Gruppen zur Vertiefung und Festigung des Lehrstoffes. Die Beurteilung erfolgt auf Basis schriftlicher Tests und kontinuierlich in Übungen und auf der Online-Plattform erbrachter Leistungen.

## **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Grundzüge digitaler Systeme

## **Statistik und Wahrscheinlichkeitstheorie**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Vermittlung der statistischen Denk- und Arbeitsweise

*Fachliche und methodische Kompetenzen:* Grundlagen der Wahrscheinlichkeitstheorie; Kenntnisse von statistischer Schätzung und statistischem Testen; Kenntnisse wichtiger statistischer Methoden

*Kognitive und praktische Kompetenzen:* Anwendung von statistischen Methodiken auf konkrete Problemstellungen; Kenntnisse im Umgang mit statistischer Software

*Soziale Kompetenzen und Selbstkompetenzen:* Umsetzung von konkreten Aufgaben in statistische Problemstellungen; Lösung statistischer Problemstellungen sowohl formal als auch mit dem Computer

**Inhalt:** Dieses Modul vermittelt im Einzelnen folgende Themen: Beschreibende Statistik, Grundlagen der Wahrscheinlichkeitstheorie, Elementare Informationstheorie, Zufallsvariablen und Verteilungen, Punkt- und Intervallschätzungen, Tests von Hypothesen, Varianzanalyse, Regression, Korrelation, Zählstatistik.

**Erwartete Vorkenntnisse:** Grundkenntnisse der Analysis und Algebra.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algebra und Diskrete Mathematik, Analysis

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Lehrveranstaltung Statistik und Wahrscheinlichkeitstheorie besteht aus einem Vorlesungsteil und einem Übungsteil. Die beschriebenen Inhalte und Konzepte werden im Rahmen der Vorlesungseinheit erläutert. Der Übungsteil besteht aus einem Teil, bei dem Beispiele analytisch gelöst werden, und einem Teil, bei dem praktische Problemstellungen mit Hilfe statistischer Software gelöst werden. Diese Veranstaltungen sollen sowohl im Winter- als auch im Sommersemester angeboten werden.

## **Lehrveranstaltungen des Moduls:**

3,0/2,0 VO Statistik und Wahrscheinlichkeitstheorie

3,0/2,0 UE Statistik und Wahrscheinlichkeitstheorie

oder

6,0/4,0 VU Statistik und Wahrscheinlichkeitstheorie

# Statistische Datenanalyse

**Regelarbeitsaufwand:** mindestens 6,0 ECTS

**Lernergebnisse:** Vermittlung einer datenorientierten und explorativen Analyse von Daten

*Fachliche und methodische Kompetenzen:*

- Grundlagen der statistischen Datenanalyse
- Detaillierte Kenntnisse und Hintergrundwissen über statistische Methodik
- Exploration und Analyse statistischer Daten
- Computergestützte Verfahren der Statistik, Schätzungen, statistische Tests, Varianzanalyse, Regression
- Kritische Sicht im Umgang und Anwendung mit statistischer Software

*Kognitive und praktische Kompetenzen:*

- Anwendung von statistischer Methodik und Software auf konkrete Problemstellungen
- Programmentwicklung mit statistischer Software
- Computerorientierte Lösung von statistischen Problemen
- Anwendung von statistischen Methodiken mit Hilfe des Computers auf konkrete Problemstellung

*Soziale Kompetenzen und Selbstkompetenzen:*

- Datenorientierte Lösung von statistischen Problemstellungen
- Lösungen von Problemen mit open-source Software

**Inhalt:** Stichprobendesign, Planung der statistischen Datenerhebung, Elemente der explorativen Datenanalyse, Grundbegriffe parametrischer/nichtparametrischer und robuster Verfahren, lineare Modelle, Einführung in multivariate statistische Methoden, Einführung in die Zeitreihenanalyse, effiziente Programmierung in der statistischen Softwareumgebung R, Datenmanipulation, statistische Graphiken, graphische Systeme in R, dynamische reports mit statistischer Software, Testen mittels statistischer Simulation. Exploration und Analyse statistischer Daten, computergestützte Verfahren der Statistik, Vertiefung in und kritischer Vergleich von kommerzieller Statistiksoftware (SPSS, SAS, SPLUS, S, R). Einsatz von spezieller Software, insbesondere Neuentwicklungen, Schwerpunkt liegt aber auf R. Arbeiten mit größeren Fallbeispielen. Der Methodenkatalog umfasst: Beschreibende Statistik, Vergleich von Gruppen von Daten, Varianzanalyse, Regressionsanalyse, Geostatistik, Ausblick in Multivariate Methoden, Simulation.

**Erwartete Vorkenntnisse:** Grundlegende Kenntnisse der Mathematik und Statistik  
Diese Voraussetzungen werden in im Modul Algebra und Diskrete Mathematik, Analysis, Statistik und Wahrscheinlichkeitstheorie vermittelt.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Lehrveranstaltung Datenanalyse vermittelt sowohl den theoretischen Zugang zu den Methoden, demonstriert aber auch die praktische Lösung mittels Statistik-Software. Die Studierenden sollen selbständig Problemstellungen mit dem Computer lösen.

Die Lehrveranstaltung Statistical Computing vermittelt einen computerorientierten Zugang zur Statistik. Ziel ist einerseits detaillierte Kenntnisse in der State-of-the-art Software R zu vermitteln, als auch statistische Probleme mittels statistischer Simulation zu lösen. Lösungen sollen durch eigenständiges Programmieren erarbeitet werden.

Die Lehrveranstaltung Computerstatistik ist ein Mix aus einem theoretischen Vorlesungsteil und einem Übungsteil, wobei die Teile in natürlicher Weise ineinander übergehen. Die theoretischen Darstellungen werden möglichst zeitnahe mit praktischen Übungen am Computer erläutert und geübt.

**Lehrveranstaltungen des Moduls:**

3,0/2,0 VU Datenanalyse

3,0/2,0 VU Statistical Computing

4,5/3,0 VU Computerstatistik

## Theoretische Informatik

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:** Dieses Modul befasst sich mit den theoretischen und logischen Grundlagen der Informatik.

*Fachliche und methodische Kompetenzen:* Fundamentale Konzepte und Resultate der Mathematischen Logik, Automaten und formalen Sprachen, Berechenbarkeit und Komplexität sowie der formalen Semantik von Programmiersprachen.

*Kognitive und praktische Kompetenzen:* Die Studierenden erwerben die Fähigkeit, formale Beschreibungen lesen und verstehen und Konzepte formal-mathematisch beschreiben zu können. Weiters lernen sie, die Struktur von Beweisen und Argumentationen zu verstehen und selber solche zu führen.

*Soziale Kompetenzen und Selbstkompetenzen:* Selbständiges Lösen von Problemen.

**Inhalt:**

- Automatentheorie: endliche Automaten, Büchautomaten, Transducer, Operationen auf Automaten
- Formale Sprachen: Chomsky Hierarchie
- Berechenbarkeit und Komplexität: universelle Berechenbarkeit, Unentscheidbarkeit, NP-Vollständigkeit
- Grundlagen der operationalen und axiomatischen Semantik von Programmiersprachen
- Grundlagen von Prozessalgebren und Concurrency (CSP, CCS)

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden in einem Vorlesungsteil vorgestellt und in begleitenden Übungen von den Studierenden erarbeitet. Die Übungsaufgaben können zeitlich und örtlich weitgehend ungebunden einzeln oder in Gruppen gelöst werden. Die Lösungen werden bei regelmäßigen Treffen mit Lehrenden und TutorInnen besprochen und korrigiert. Die Beurteilung erfolgt auf Basis schriftlicher Tests und der kontinuierlich in den Übungen erbrachten Leistungen. Der Übungsbetrieb und die Tests können computerunterstützt durchgeführt werden.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Theoretische Informatik

## Übersetzerbau

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls verstehen die Studierenden die theoretischen Grundlagen des Übersetzerbaus und können diese erklären und anwenden.

*Kognitive und praktische Kompetenzen:* Durch die Auseinandersetzung mit Methoden und Werkzeugen des Übersetzerbaus erwerben die Studierenden

- die praktische Fähigkeit zur Assemblerprogrammierung und
- die praktische Fähigkeit zur Konstruktion von Parsern und Übersetzern

*Soziale Kompetenzen und Selbstkompetenzen:*

- Neugierde am Übersetzerbau

**Inhalt:**

- Grundlagen von Übersetzern und Interpretern, Struktur von Übersetzern
- Computerarchitektur
- lexikalische Analyse (reguläre Definition, endlicher Automat)
- Syntax-Analyse (Top-Down, Bottom-Up)
- syntaxgesteuerte Übersetzung (attributierte Grammatik)
- semantische Analyse, Zwischencode (Symboltabelle)
- Zwischendarstellungen
- Codeerzeugung (Befehlsauswahl, Befehlsanordnung, Registerbelegung)
- Laufzeitsystem (Stackverwaltung, Heapverwaltung)
- Optimierungen (Programmanalysen, skalare Optimierungen, Schleifenoptimierungen)

- Übersetzung objektorientierter Konzepte (Klassendarstellung und Methodenaufruf, Typüberprüfung, Analysen)

**Erwartete Vorkenntnisse:**

- theoretische Grundlagen der Informatik
- alle zur Erstellung von Programmen notwendigen Kenntnisse
- die praktische Fähigkeit zur Konstruktion von Programmen

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Algorithmen und Datenstrukturen, Einführung in die Programmierung, Programmierparadigmen

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Vortrag und selbständiges Erlernen der eher theoretischen Grundlagen. Laborübung in geführten Kleingruppen zur Entwicklung praktischer Übersetzerentwicklungsfähigkeiten. Die Beurteilung erfolgt durch Prüfung oder Tests und die Beurteilung der Lösungen von Programmieraufgaben plus Abschlussgespräch.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Übersetzerbau

**Usability Engineering and Mobile Interaction**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Dieses Modul befasst sich mit der User-Research getriebenen Konzeption, Gestaltung und Evaluierung von Benutzerschnittstellen. Das Modul gliedert sich in zwei große Teilgebiete: Im ersten Teil werden die grundlegenden Konzepte von Usability Engineering gelehrt. Anhand von praxisnahen Beispielen sollen Studierende den Einsatz von Usability Engineering erlernen. Die gelehrt Methoden decken den gesamten Design Prozess von Requirements Engineering (z.B. Kontextuelle Interviews), Prototyping bis hin zum Testen von Systemen (z.B. Usability Test, Heuristische Evaluierung) ab. Der zweite Teil dieses Moduls ist den Methoden der Mobile Interaction Research gewidmet, mit besonderem Fokus auf aktuellen Entwicklungen und Trends. Aufbauend auf den Grundlagen des Usability Engineerings werden Besonderheiten und Spezifika sowohl im Design als auch in der Evaluierung von mobilen Anwendungen hervorgehoben.

*Kognitive und praktische Kompetenzen:* Das Modul vermittelt Kenntnisse über Qualitätskriterien für gute Usability sowie deren Evaluierung und Beurteilung anhand etablierter Usability Engineering Methoden und zeigt aktuelle Entwicklungen und zukünftige Trends im Bereich der Mobile Interaction auf.

*Soziale Kompetenzen und Selbstkompetenzen:* Das Modul vermittelt die Bedeutung von Usability Engineering für den Erfolg von Softwareentwicklungsprojekten und geht auf

die Möglichkeiten und Herausforderungen der Einbindung von Usability Engineering Methoden in Software Engineering Prozessen ein.

**Inhalt:** Usability Engineering:

- Einführung in Usability Engineering
- Qualitätskriterien für Usability Engineering und deren Messung und Beurteilung
- Usability Engineering Lifecycle
- Methoden des Usability Engineerings in Anlehnung an die Phasen des Human Centered Design Prozesses: Kontextanalyse, Requirementsanalyse, Design and Prototyping, Evaluierung
- Praktische Anwendung der vorgestellten Methoden in einem Übungsteil

Pilots in Mobile Interaction: User-centered Interaction Research and Evaluation:

- Einführung in User-centered Interaction Research
- Quality of Experience Methods and Applications
- Perceptual Quality for Mediated Interaction
- Cognitive User Interfaces
- Audio-Visual Speech Synthesis
- Advanced Mobile Spatial Interaction
- Rapid Prototyping for Future Mobile Interactions
- Case Study of Mobile Interfaces in Large IT Projects

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** In Form von Vorlesungen mit Übungen werden die vertiefenden Inhalte vermittelt. In der Vorlesung werden theoretische Grundlagen vermittelt. Die in der Vorlesung vorgestellten Methoden sind in einem praktischen Übungsteil umzusetzen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Usability Engineering and Mobile Interaction

## **Verteilte Systeme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Anforderungen und Designmöglichkeiten komplexer, verteilter Systeme verstehen
- Grundlegende Methoden und Algorithmen verteilter Systeme verstehen, sowie deren Vor- und Nachteile und Einsatzmöglichkeiten kennen
- Paradigmen und Konzepte aktueller Technologien und Werkzeuge für verteilte Systeme verstehen und anwenden können
- Anwendungsgrenzen (v. a. asynchroner) verteilter Systeme kennen und verstehen

*Kognitive und praktische Kompetenzen:* Durch die Auseinandersetzung mit Methoden und Werkzeugen der Programmierung können die Studierenden

- Methodiken zur Abstraktion anwenden,
- methodisch fundiert an Probleme herangehen,
- Lösungen kritisch bewerten und reflektieren und
- Konzepte verteilter Systeme mit aktuellen Technologien in Form einfacher, verteilter Anwendungen umsetzen.

*Soziale Kompetenzen und Selbstkompetenzen:* Folgende Kompetenzen werden besonders gefördert:

- Selbstorganisation und Eigenverantwortlichkeit
- Finden kreativer Problemlösungen
- Kritische Reflexion, Bewertung und Analyse technischer Alternativansätze

### **Inhalt:**

- Verteilte Systeme – Übersicht, Grundlagen und Modelle
- Prozesse und Kommunikation
- Benennung
- Fehlertoleranz in verteilten Systemen
- Synchronisierung
- Konsistenz und Replikation
- Verteilte Dateisysteme
- Sicherheit
- Anwendungen und Technologietrends

### **Erwartete Vorkenntnisse:**

Dieses Modul baut auf den Kenntnissen, Fertigkeiten und Kompetenzen folgender Module auf: Algorithmen und Datenstrukturen, Einführung in die Programmierung, Programmierparadigmen

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:**  
Blended Learning:

- Den Studierenden wird empfohlen, vor der jeweiligen Vorlesung die auf der LVA-Homepage angegebenen Kapitel des Lehrbuchs zu lesen.
- Im Rahmen der Vorlesung wird die Theorie erläutert und Querverbindungen hergestellt. Es besteht die Möglichkeit, komplexe Sachverhalte interaktiv (durch Fragen der Studierenden) zu erarbeiten.
- Im Rahmen der parallel laufenden Laborübungen werden ausgewählte Themen der Lehrveranstaltung durch kleine Programmieraufgaben vertieft.

### **Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Verteilte Systeme

# Web Engineering

**Regelarbeitsaufwand:** 6,0 ECTS

## **Lernergebnisse:**

*Fachliche und methodische Kompetenzen:*

- Das Modul vermittelt ein breites und integriertes Wissen und Verstehen der wissenschaftlichen Grundlagen der Entwicklung von Web-Anwendungen, welches wesentlich über das auf der Ebene der Universitätszugangsberechtigung vorhandene Wissen hinausgeht.
- Die Studierenden verfügen über ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Konzepte zur Entwicklung von Web-Anwendungen.
- Das Wissen und Verständnis der Studierenden entspricht dem Stand der Fachliteratur in diesem Bereich.

*Kognitive und praktische Kompetenzen:*

- Die Studierenden können ihr Wissen und Verstehen praktisch zur Lösung von Aufgaben im Bereich der Web-Anwendungsentwicklung umsetzen.
- Die Studierenden können für die jeweilig vorliegende Aufgabenstellung erkennen welche Abstraktionen zu verwendet oder entwickelt werden müssen.

*Soziale Kompetenzen und Selbstkompetenzen:*

- Die Studierenden sind in der Lage ihr Wissen selbständig zu vertiefen.
- Die Studierenden können entwicklungsbezogene Positionen und Problemlösungen formulieren, sich mit InformatikerInnen und DomänenexpertInnen darüber austauschen und Verantwortung in einem Team übernehmen.
- Die Studierenden lernen ihre eigenen Fähigkeiten und Grenzen einzuschätzen und erwerben die Kritikfähigkeit an der eigenen Arbeit.
- Die Studierenden erlernen Selbstorganisation und Eigenverantwortlichkeit zum eigenständigen Lösen von Aufgaben.

## **Inhalt:**

- Grundlagen: Begriffsdefinitionen; Strukturelle Grundlagen des World Wide Web (Protokolle, Web Server Infrastrukturen); Grundlagen von Web-Anwendungen.
- Sprachen und Technologien: Grundlagen von Markup-Sprachen und Stylesheet Sprachen; Technologien für interaktive und dynamische Web-Anwendungen im Browser; Komponenten-basierte Entwicklung; Web Services.
- Umsetzung und praktische Realisierung: Barrierefreie Web-Anwendungen; Entwicklungsprozess; Entwurfsmuster; Entwicklungs-APIs.

**Erwartete Vorkenntnisse:**

Es werden die Fertigkeiten und Kenntnisse folgender Module benötigt: Einführung in die Programmierung.

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Inhalte werden durch Vortragseinheiten vermittelt und durch theoretische und praktische Übungen geübt. Die Beurteilung basiert auf Tests über den vorgetragenen Stoff sowie auf den erbrachten Leistungen in den Übungen.

**Lehrveranstaltungen des Moduls:**

6,0/4,0 VU Web Engineering

**Zuverlässige Echtzeitsysteme**

**Regelarbeitsaufwand:** 6,0 ECTS

**Lernergebnisse:**

*Fachliche und methodische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Methoden zur Konstruktion und Modellierung von zuverlässigen Systemen, die strikten zeitlichen Vorgaben gehorchen müssen, beschreiben und anwenden. Dazu benutzen sie die erworbenen Kenntnisse über Fehlerarten, Fehlermodelle, Fehlererkennung, Fehleranalyse, Redundanzverfahren, Zuverlässigkeitsmodellierung.
- Konzepte und Methoden der zeitabhängigen Information, Uhrensynchronisation, Echtzeitscheduling und Echtzeitkommunikation erklären und benutzen.
- Risiken des Einsatzes von Computersystemen in sicherheitskritischen und zeitkritischen Anwendungen diskutieren und beurteilen.

*Kognitive und praktische Kompetenzen:* Nach positiver Absolvierung des Moduls können die Studierenden:

- Fehlerwahrscheinlichkeiten und Ausfallrisiken einschätzen und modellieren,
- Verfahren zur Erhöhung der Zuverlässigkeit von Computersystemen anwenden,
- zeitliche Anforderungen an Computersysteme analysieren,
- Computersysteme mit Echtzeitanforderungen entwerfen und modellieren.

**Inhalt:**

- Grundlagen: Zuverlässigkeit, Wartbarkeit, Verfügbarkeit, MTTF
- Quantitative Analysen: Blockdiagramme, Fehlerbäume, Markov-Prozesse
- Sicherheit, Fehlermodelle, Wartung, Alterungsfehler, Entwurfsfehler
- Fehlertolerante Computersysteme: Redundanz, Fehlerlatenz, Synchronisation, Voting, Recovery Blocks, N-Version-Programming
- Fallstudien von zuverlässigen bzw. fehlertoleranten Systemen

- Fehler und Zuverlässigkeitsmodellierung/analyse mit Tools
- Grundlagen Echtzeitsysteme, Zeitabhängigkeit von Information, logische und temporale Ordnung
- Modellbildung von Echtzeitsystemen: Zustand und Ereignis, Komponenten, Interfaces, Echtzeitinformation
- Echtzeitkommunikation, Kommunikationsprotokolle für Echtzeitsysteme
- Uhrensynchronisation
- Fehlertoleranz in Echtzeitsystemen
- Echtzeitbetriebssysteme: Taskstruktur, Ressourcenmanagement, I/O, Scheduling, Worst-Case Zeitanalyse von Tasks
- Energieverbrauch und Energiemanagement in Echtzeitsystemen
- Design von Echtzeitsystemen: Architekturmodelle, Composability, Designprinzipien, Zertifizierung

### **Erwartete Vorkenntnisse:**

*Fachliche und methodische Kompetenzen:* Kenntnisse in Boole'scher Algebra, theoretischer Informatik und Logik, Wahrscheinlichkeitsrechnung, Beschreibung und Modellierung stochastischer Prozesse, Aufbau und Funktionsweise von Microcomputern, Betriebssystemen und Netzwerken.

*Kognitive und praktische Kompetenzen:* Wahrscheinlichkeitsrechnung und Modellierung, Systematisches Arbeiten mit Softwaretools, Abstraktionsvermögen.

*Soziale Kompetenzen und Selbstkompetenzen:* Analyse komplexer Zusammenhänge und Wechselwirkungen, Problemlösung im Team.

Diese Voraussetzungen werden in folgenden Modulen vermittelt: Grundzüge digitaler Systeme, Theoretische Informatik, Statistik und Wahrscheinlichkeitstheorie, Betriebssysteme

**Verpflichtende Voraussetzungen:** Studieneingangs- und Orientierungsphase.

**Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung:** Die Grundlagen und theoretischen Inhalte werden im Vorlesungsteil vermittelt. Praktische Fertigkeiten der Fehler- und Zuverlässigkeitsmodellierung werden in einem Übungsteil erworben, in dem die Studierenden Softwaretools verwenden und Ergebnisse in Form von Laborberichten dokumentieren.

### **Lehrveranstaltungen des Moduls:**

3,0/2,0 VO Echtzeitsysteme

3,0/2,0 VU Dependable Systems

## B. Lehrveranstaltungstypen

**EX:** Exkursionen sind Lehrveranstaltungen, die außerhalb des Studienortes stattfinden. Sie dienen der Vertiefung von Lehrinhalten im jeweiligen lokalen Kontext.

**LU:** Laborübungen sind Lehrveranstaltungen, in denen Studierende in Gruppen unter Anleitung von Betreuer\_innen experimentelle Aufgaben lösen, um den Umgang mit Geräten und Materialien sowie die experimentelle Methodik des Faches zu lernen. Die experimentellen Einrichtungen und Arbeitsplätze werden zur Verfügung gestellt.

**PR:** Projekte sind Lehrveranstaltungen, in denen das Verständnis von Teilgebieten eines Faches durch die Lösung von konkreten experimentellen, numerischen, theoretischen oder künstlerischen Aufgaben vertieft und ergänzt wird. Projekte orientieren sich an den praktisch-beruflichen oder wissenschaftlichen Zielen des Studiums und ergänzen die Berufsvorbildung bzw. wissenschaftliche Ausbildung.

**SE:** Seminare sind Lehrveranstaltungen, bei denen sich Studierende mit einem gestellten Thema oder Projekt auseinandersetzen und dieses mit wissenschaftlichen Methoden bearbeiten, wobei eine Reflexion über die Problemlösung sowie ein wissenschaftlicher Diskurs gefordert werden.

**UE:** Übungen sind Lehrveranstaltungen, in denen die Studierenden das Verständnis des Stoffes der zugehörigen Vorlesung durch Anwendung auf konkrete Aufgaben und durch Diskussion vertiefen. Entsprechende Aufgaben sind durch die Studierenden einzeln oder in Gruppenarbeit unter fachlicher Anleitung und Betreuung durch die Lehrenden (Universitätslehrer\_innen sowie Tutor\_innen) zu lösen. Übungen können auch mit Computerunterstützung durchgeführt werden.

**VO:** Vorlesungen sind Lehrveranstaltungen, in denen die Inhalte und Methoden eines Faches unter besonderer Berücksichtigung seiner spezifischen Fragestellungen, Begriffsbildungen und Lösungsansätze vorgetragen werden. Bei Vorlesungen herrscht keine Anwesenheitspflicht.

**VU:** Vorlesungen mit integrierter Übung vereinen die Charakteristika der Lehrveranstaltungstypen VO und UE in einer einzigen Lehrveranstaltung.

## **C. Zusammenfassung aller verpflichtenden Voraussetzungen**

Die positiv absolvierte Studieneingangs- und Orientierungsphase (Abschnitt 7) ist Voraussetzung für die Absolvierung aller in diesem Studienplan angeführten Module und ihrer Lehrveranstaltungen (inklusive der Bachelorarbeit), ausgenommen die Module

Algebra und Diskrete Mathematik (9,0 ECTS)

Algorithmen und Datenstrukturen (8,0 ECTS)

Analysis (6,0 ECTS)

Denkweisen der Informatik (6,5 ECTS)

Datenbanksysteme (6,0 ECTS)

Einführung in die Programmierung (9,5 ECTS)

Einführung in Visual Computing (6,0 ECTS)

Grundzüge digitaler Systeme (6,0 ECTS)

## D. Semestereinteilung der Lehrveranstaltungen

Die in der nachfolgenden Semestereinteilung mit Stern markierten Lehrveranstaltungen setzen eine positiv absolvierte Studieneingangs- und Orientierungsphase voraus.

### 1. Semester (WS)

4,0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
5,0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
5,5 VU Denkweisen der Informatik  
5,5 VU Einführung in die Programmierung 1  
6,0 VU Grundzüge digitaler Systeme  
1,0 VU Orientierung Informatik und Wirtschaftsinformatik

### 2. Semester (SS)

8,0 VU Algorithmen und Datenstrukturen  
2,0 VO Analysis für Informatik und Wirtschaftsinformatik  
4,0 UE Analysis für Informatik und Wirtschaftsinformatik  
6,0 VU Datenbanksysteme  
4,0 VU Einführung in die Programmierung 2  
6,0 VU Einführung in Visual Computing

### 3. Semester (WS)

\* 6,0 VU Betriebssysteme  
\* 6,0 VU Programmierparadigmen  
\* 3,0 VO Statistik und Wahrscheinlichkeitstheorie  
\* 3,0 UE Statistik und Wahrscheinlichkeitstheorie  
\* 6,0 VU Software Engineering  
\* 6,0 VU Theoretische Informatik

### 4. Semester (SS)

\* 6,0 VU Daten- und Informatikrecht  
\* 6,0 VU Einführung in paralleles Rechnen (Parallel Computing)  
\* 6,0 PR Software Engineering Projekt

### 5. Semester (WS)

\* 6,0 VU Einführung in Artificial Intelligence  
\* 6,0 VU Interface und Interaction Design  
\* 6,0 VU Einführung in Security

- \* 6,0 VU Verteilte Systeme
- \* 3,0 SE Wissenschaftliches Arbeiten

## **6. Semester (SS)**

- \*10,0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik

## **E. Semesterempfehlung für schiefeinsteigende Studierende**

Bei Beginn des Studiums im Sommersemester ist zu beachten, dass die in der nachfolgenden Semestereinteilung mit Stern markierten Lehrveranstaltungen eine positiv absolvierte Studieneingangs- und Orientierungsphase voraussetzen. Daher ist ein schiefe-mestriger Einstieg nur jenen Studierenden anzuraten, die in der Lage sind, sämtliche Lehrveranstaltungen des ersten Semesters bis zum Beginn des zweiten Semesters positiv abzuschließen.

### **1. Semester (SS)**

4,0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
5,0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik  
2,0 VO Analysis für Informatik und Wirtschaftsinformatik  
4,0 UE Analysis für Informatik und Wirtschaftsinformatik  
5,5 VU Einführung in die Programmierung 1  
1,0 VU Orientierung Informatik und Wirtschaftsinformatik

### **2. Semester (WS)**

5,5 VU Denkweisen der Informatik  
6,0 VU Grundzüge digitaler Systeme  
\* 6,0 VU Einführung in Security  
\* 6,0 VU Programmierparadigmen

### **3. Semester (SS)**

8,0 VU Algorithmen und Datenstrukturen  
6,0 VU Daten- und Informatikrecht  
6,0 VU Datenbanksysteme  
4,0 VU Einführung in die Programmierung 2  
6,0 VU Einführung in Visual Computing

### **4. Semester (WS)**

\* 6,0 VU Betriebssysteme  
\* 3,0 VO Statistik und Wahrscheinlichkeitstheorie  
\* 3,0 UE Statistik und Wahrscheinlichkeitstheorie  
\* 6,0 VU Software Engineering  
\* 6,0 VU Theoretische Informatik  
\* 6,0 VU Verteilte Systeme

## **5. Semester (SS)**

- \* 6,0 PR Software Engineering Projekt
- \* 3,0 SE Wissenschaftliches Arbeiten

## **6. Semester (WS)**

- \* 6,0 VU Einführung in Artificial Intelligence
- \* 6,0 VU Interface und Interaction Design
- \* 6,0 VU Einführung in paralleles Rechnen (Parallel Computing)
- \*10,0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik



1. Semester (27 ECTS)	5,5 VU Einführung in die Programmierung 1	4,0 VO / 5,0 UE Algebra und Diskrete Mathematik	5,5 VU Denkweisen der Informatik	1,0 VU Orientierung Informatik	6,0 VU Grundzüge digitaler Systeme
2. Semester (30 ECTS)	4,0 VU Einführung in die Programmierung 2	2,0 VO / 4,0 UE Analysis	8,0 VU Algorithmen und Datenstrukturen	6,0 VU Datenbanksysteme	6,0 VU Einführung in Visual Computing
3. Semester (30 ECTS)	6,0 VU Programmierparadigmen	3,0 VO / 3,0 UE Statistik und Wahrscheinlichkeitstheorie	6,0 VU Betriebsysteme	6,0 VU Theoretische Informatik	6,0 VU Software Engineering
4. Semester (18 ECTS)	6,0 VU Parallel Computing	6,0 VU Daten- und Informatikrecht	6,0 VU Software Engineering Projekt		
5. Semester (27 ECTS)	6,0 VU Einführung in Artificial Intelligence	6,0 VU Einführung in Security	6,0 VU Verteilte Systeme	6,0 VU Interface und Interaction Design	3,0 SE Wissenschaftliches Arbeiten

## F. Wahlfachkatalog „Transferable Skills“

Die Lehrveranstaltungen, die im Modul *Freie Wahlfächer und Transferable Skills* aus dem Themenbereich „Transferable Skills“ zu wählen sind, können unter anderem aus dem folgenden Katalog gewählt werden.

- 3,0/2,0 SE Coaching als Führungsinstrument 1
- 3,0/2,0 SE Coaching als Führungsinstrument 2
- 3,0/2,0 SE Didaktik in der Informatik
- 1,5/1,0 VO EDV-Vertragsrecht
- 3,0/2,0 VO Einführung in die Wissenschaftstheorie I
- 3,0/2,0 VO Einführung in Technik und Gesellschaft
- 3,0/2,0 SE Folgenabschätzung von Informationstechnologien
- 3,0/2,0 VU Forschungsmethoden
- 3,0/2,0 VO Frauen in Naturwissenschaft und Technik
- 3,0/2,0 SE Gruppendynamik
- 3,0/2,0 VU Kommunikation und Moderation
- 3,0/2,0 SE Kommunikation und Rhetorik
- 1,5/1,0 SE Kommunikationstechnik
- 3,0/2,0 VU Kooperatives Arbeiten
- 3,0/2,0 VU Präsentation und Moderation
- 1,5/1,0 VO Präsentation, Moderation und Mediation
- 3,0/2,0 UE Präsentation, Moderation und Mediation
- 3,0/2,0 VU Präsentations- und Verhandlungstechnik
- 4,0/4,0 SE Privatissimum aus Fachdidaktik Informatik
- 3,0/2,0 VU Rhetorik, Körpersprache, Argumentationstraining
- 3,0/2,0 VU Softskills für TechnikerInnen
- 3,0/2,0 VU Techniksoziologie und Technikpsychologie
- 3,0/2,0 VO Theorie und Praxis der Gruppenarbeit
- 3,0/2,0 VO Zwischen Karriere und Barriere

## **G. Prüfungsfächer mit den zugeordneten Modulen und Lehrveranstaltungen**

Die mit einem Stern markierten Module sind Wahl-, die übrigen Pflichtmodule.

### **Prüfungsfach „Algorithmen und Programmierung“**

#### **Modul „Algorithmen und Datenstrukturen“ (8,0 ECTS)**

8,0/5,5 VU Algorithmen und Datenstrukturen

#### **Modul „Einführung in die Programmierung“ (9,5 ECTS)**

5,5/4,0 VU Einführung in die Programmierung 1

4,0/3,0 VU Einführung in die Programmierung 2

#### **Modul „Einführung in paralleles Rechnen (Parallel Computing)“ (6,0 ECTS)**

6,0/4,0 VU Einführung in paralleles Rechnen (Parallel Computing)

#### **Modul „Programmierparadigmen“ (6,0 ECTS)**

6,0/4,0 VU Programmierparadigmen

#### **\*Modul „Deklaratives Problemlösen“ (6,0 ECTS)**

6,0/4,0 VU Deklaratives Problemlösen

#### **\*Modul „Funktionale Programmierung“ (6,0 ECTS)**

6,0/4,0 VU Funktionale Programmierung

#### **\*Modul „Logikprogrammierung und Constraints“ (6,0 ECTS)**

6,0/4,0 VU Logikprogrammierung und Constraints

### **Prüfungsfach „Computersysteme“**

#### **Modul „Betriebssysteme“ (6,0 ECTS)**

6,0/4,0 VU Betriebssysteme

#### **Modul „Einführung in Visual Computing“ (6,0 ECTS)**

6,0/4,0 VU Einführung in Visual Computing

#### **Modul „Grundzüge digitaler Systeme“ (6,0 ECTS)**

6,0/4,0 VU Grundzüge digitaler Systeme

#### **Modul „Verteilte Systeme“ (6,0 ECTS)**

6,0/4,0 VU Verteilte Systeme

#### **\*Modul „Abstrakte Maschinen“ (6,0 ECTS)**

6,0/4,0 VU Abstrakte Maschinen

**\*Modul „Übersetzerbau“ (6,0 ECTS)**

6,0/4,0 VU Übersetzerbau

**\*Modul „Grundlagen der Computergraphik“ (6,0 ECTS)**

6,0/4,0 VU Grundlagen der Computergraphik

**\*Modul „Zuverlässige Echtzeitsysteme“ (6,0 ECTS)**

3,0/2,0 VO Echtzeitsysteme

3,0/2,0 VU Dependable Systems

**Prüfungsfach „Informatik und Gesellschaft“**

**Modul „Denkweisen der Informatik“ (6,5 ECTS)**

5,5/4,0 VU Denkweisen der Informatik

1,0/1,0 VU Orientierung Informatik und Wirtschaftsinformatik

**Modul „Interface und Interaction Design“ (6,0 ECTS)**

6,0/4,0 VU Interface und Interaction Design

**Modul „Einführung in Security“ (6,0 ECTS)**

6,0/4,0 VU Einführung in Security

**Modul „Daten- und Informatikrecht“ (mindestens 6,0 ECTS)**

6,0/4,0 VU Daten- und Informatikrecht

3,0/3,0 VU Vertrags- und Haftungsrecht

**\*Modul „Security“ (mindestens 6,0 ECTS)**

6,0/4,0 VU Introduction to Cryptography

6,0/4,0 VU Privacy Enhancing Technologies

3,0/2,0 VU Security for Systems Engineering

3,0/2,0 VU Mobile Security

**Prüfungsfach „Information Engineering“**

**Modul „Datenbanksysteme“ (6,0 ECTS)**

6,0/4,0 VU Datenbanksysteme

**Modul „Einführung in Artificial Intelligence“ (6,0 ECTS)**

6,0/4,0 VU Einführung in Artificial Intelligence

**\*Modul „Einführung in wissensbasierte Systeme“ (6,0 ECTS)**

6,0/4,0 VU Einführung in wissensbasierte Systeme

**\*Modul „Logik für Wissensrepräsentation“ (6,0 ECTS)**

6,0/4,0 VU Logik für Wissensrepräsentation

**\*Modul „Semistrukturierte Daten“ (6,0 ECTS)**

6,0/4,0 VU Semistrukturierte Daten

**\*Modul „Statistische Datenanalyse“ (mindestens 6,0 ECTS)**

3,0/2,0 VU Datenanalyse

3,0/2,0 VU Statistical Computing

4,5/3,0 VU Computerstatistik

**\*Modul „Web Engineering“ (6,0 ECTS)**

6,0/4,0 VU Web Engineering

**Prüfungsfach „Mathematik, Statistik und Theoretische Informatik“**

**Modul „Algebra und Diskrete Mathematik“ (9,0 ECTS)**

4,0/4,0 VO Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

5,0/2,0 UE Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

9,0/6,0 VU Algebra und Diskrete Mathematik für Informatik und Wirtschaftsinformatik

**Modul „Analysis“ (6,0 ECTS)**

2,0/2,0 VO Analysis für Informatik und Wirtschaftsinformatik

4,0/2,0 UE Analysis für Informatik und Wirtschaftsinformatik

6,0/4,0 VU Analysis für Informatik und Wirtschaftsinformatik

**Modul „Statistik und Wahrscheinlichkeitstheorie“ (6,0 ECTS)**

3,0/2,0 VO Statistik und Wahrscheinlichkeitstheorie

3,0/2,0 UE Statistik und Wahrscheinlichkeitstheorie

6,0/4,0 VU Statistik und Wahrscheinlichkeitstheorie

**Modul „Theoretische Informatik“ (6,0 ECTS)**

6,0/4,0 VU Theoretische Informatik

**\*Modul „Argumentieren und Beweisen“ (6,0 ECTS)**

6,0/4,0 VU Argumentieren und Beweisen

**\*Modul „Introduction to Numerics“ (4,5 ECTS)**

4,5/3,0 VU Introduction to Numerics

**\*Modul „Logic and Reasoning in Computer Science“ (6,0 ECTS)**

6,0/4,0 VU Logic and Reasoning in Computer Science

**\*Modul „Multivariate und computerintensive statistische Methoden“ (9,0 ECTS)**

4,5/3,0 VO Multivariate Statistik

1,5/1,0 UE Multivariate Statistik

3,0/2,0 VU Statistische Simulation und computerintensive Methoden

**Prüfungsfach „Software Engineering“**

**Modul „Software Engineering“ (6,0 ECTS)**

6,0/4,0 VU Software Engineering

**Modul „Software Engineering Projekt“ (6,0 ECTS)**

6,0/4,0 PR Software Engineering Projekt

**\*Modul „Programm- und Systemverifikation“ (6,0 ECTS)**

6,0/4,0 VU Programm- und Systemverifikation

**\*Modul „Softwareprojekt-Beobachtung und -Controlling“ (6,0 ECTS)**

6,0/4,0 VU Softwareprojekt-Beobachtung und -Controlling

**\*Modul „Software-Qualitätssicherung“ (6,0 ECTS)**

6,0/4,0 VU Software-Qualitätssicherung

**\*Modul „Usability Engineering and Mobile Interaction“ (6,0 ECTS)**

6,0/4,0 VU Usability Engineering and Mobile Interaction

**Prüfungsfach „Freie Wahlfächer und Transferable Skills“**

**Modul „Freie Wahlfächer und Transferable Skills“ (18,0 ECTS)**

**Prüfungsfach „Bachelorarbeit“**

**Modul „Bachelorarbeit“ (13,0 ECTS)**

10,0/5,0 PR Bachelorarbeit für Informatik und Wirtschaftsinformatik

3,0/2,0 SE Wissenschaftliches Arbeiten

## H. Bachelor-Abschluss with Honors

Als Erweiterung eines regulären Bachelor-Studien der Informatik können Studierende mit hervorragenden Studienleistungen einen *Bachelor-Abschluss with Honors* nach anglo-amerikanischem Vorbild erwerben.

Die primären Ziele des Honors-Programms der Informatik und der Wirtschaftsinformatik sind:

- Individuelle Förderung und Forderung besonders begabter Studierender.
- Frühzeitige Erweckung des Forschungsinteresses in potentiellen Kandidatinnen und Kandidaten für ein späteres Doktoratsstudium.
- Erhöhung der Attraktivität der TU Wien und der Fakultät für Informatik für hervorragende Studieninteressierte.

Notwendige Bedingung für den *Bachelor-Abschluss with Honors* sind 45 bis 60 ECTS an zusätzlichen Bachelor- und/oder Master-Lehrveranstaltungen. Das jeweilige *individuelle Honors-Programm* wird von dem/der Studierenden in Abstimmung mit einem als Mentor/-in agierenden habilitierten Mitglied der Fakultät für Informatik individuell zusammengestellt und beim zuständigen studienrechtlichen Organ eingereicht. Die Lehrveranstaltungen des individuellen Honors-Programms sollen vorrangig so ausgewählt werden, dass sie auch in einem parallelen anderen Bachelor- oder einem anschließenden Master-Studium verwendet werden können.

Für den erfolgreichen *Bachelor-Abschluss with Honors* ist es erforderlich, das Bachelorstudium mit Auszeichnung<sup>1</sup> und sowohl alle für den Abschluss dieses Bachelorstudiums erforderlichen Lehrveranstaltungen mit einem gewichteten Gesamtnotenschnitt  $\leq 1,5$  als auch in Summe alle für den Abschluss dieses Bachelorstudiums erforderlichen Lehrveranstaltungen und jene im Rahmen des individuellen Honors-Programms absolvierten Lehrveranstaltungen mit einem gewichteten Gesamtnotenschnitt  $\leq 1,5$  innerhalb von maximal 9 Semestern zu absolvieren (gegebenenfalls unter angemessener Berücksichtigung von Beurlaubung und Teilzeit). Als Bestätigung für den *Bachelor-Abschluss with Honors* wird vom Rektorat der TU Wien ein Zertifikat ausgestellt, das die hervorragenden Studienleistungen bestätigt und die im Rahmen des individuellen Honors-Programms absolvierten zusätzlichen Lehrveranstaltungen anführt.

### H.1. Antragstellung und Aufnahme in das Honors-Programm

Nach positiver Absolvierung von mindestens 72 ECTS an Pflichtlehrveranstaltungen des Bachelorstudiums kann von der/dem Studierenden, in Abstimmung mit einem als Mentor/in agierenden habilitierten Mitglied der Fakultät für Informatik, das individuelle Honors-Programm zusammengestellt und zusammen mit einem Nachweis über die bisherigen Studienleistungen, d.h. über die für das gegenständliche Bachelorstudium absolvierten Lehrveranstaltungen, beim zuständigen studienrechtlichen Organ als Antrag auf Aufnahme in das Honors-Programm der Informatik und Wirtschaftsinformatik eingereicht werden. Das individuelle Honors-Programm muss auch ein kurze Rechtfertigung

---

<sup>1</sup>im Sinne des Par. 73 Abs. 3 UG in der Fassung vom 26. Juni 2017

(„Qualifikationsprofil“) für die getroffene Auswahl der Lehrveranstaltungen enthalten. Darüber hinaus kann jede\_r Studierende\_r auch ohne die Erfüllung dieser Eingangsvoraussetzungen einen Antrag auf Aufnahme in das Programm für einen Bachelor-Abschluss with Honors stellen, wenn diese\_r Studierende Empfehlungsschreiben von zwei habilitierten Personen (eine davon auch als Mentor\_in) vorlegen kann. Das studienrechtliche Organ entscheidet nach qualitativer Prüfung des bisherigen Studienfortschritts über die Aufnahme.

Die konkreten Lehrveranstaltungen des individuellen Honors-Programms können beliebig aus Informatik-vertiefenden oder ergänzenden Pflichtlehrveranstaltungen aus universitären Bachelor-Studien und Pflicht- oder Wahllehrveranstaltungen aus universitären Masterstudien gewählt werden, unter Beachtung der gegebenenfalls erforderlichen Vorkenntnisse. Die Lehrveranstaltungen des individuelle Honors-Programms sollen vorrangig so ausgewählt werden, dass sie auch in einem parallelen anderen Bachelor- oder einem anschließenden Master-Studium verwendet werden können. Jedenfalls zu wählen ist die spezielle Lehrveranstaltung

1,0/1,0 VU Mentoring für das Honors-Programm

1,0/1,0 VU Mentoring für das Honors-Programm  
die das individuelle Mentoring abdeckt.

Das studienrechtliche Organ überprüft folgende Bedingungen zur Aufnahme in das Honors-Programm der Informatik und Wirtschaftsinformatik:

- (a) Der/Die Studierende hat Pflichtlehrveranstaltungen des Bachelorstudiums im Ausmaß von mindestens 72 ECTS positiv absolviert.
- (b) Der gewichtete Notenschnitt aller bis zum Zeitpunkt der Antragstellung für den Abschluss des regulären Bachelor-Studiums absolvierten Lehrveranstaltungen muss  $\leq 2,0$  sein.
- (c) Ein adäquates, alle Lehrveranstaltungsabhängigkeiten berücksichtigendes individuelles Honors-Programm liegt vor.
- (d) Allfällige Kapazitätslimits (z.B. der Betreuungskapazität der Mentorin/des Mentors) werden nicht überschritten.
- (e) Der/Die Studierende muss auf Basis der bisher erbrachten Leistungen, unter der Annahme eines zumutbaren Studienfortschritts, die Bedingungen für einen erfolgreichen *Bachelor-Abschluss with Honors* erfüllen können.
- (f) Bei einem Antrag auf Aufnahme in das Programm auf Basis von Empfehlungsschreiben von zwei habilitierten Personen müssen die Kriterien (a) und (b) nicht erfüllt sein. Die Erfüllung dieser Kriterien wird in diesem Fall durch eine qualitative Prüfung des Studienfortschritts durch das studienrechtliche Organ ersetzt.

Nach positivem Bescheid über die Aufnahme in das Honors-Programm der Informatik und Wirtschaftsinformatik verbleibt die/der Studierende bis zum erfolgreichen Abschluss

oder bis zu einem eventuellen vorzeitigen Ausstieg (wie Abmeldung oder Studienwechsel), höchstens aber für 9 Semester in diesem Programm. Ein Abschluss des Bachelorstudiums ist zwischenzeitlich möglich<sup>2</sup>, ohne dass davon das Recht auf einen späteren *Bachelor-Abschluss with Honors* berührt würde, wenn schlussendlich alle notwendigen Kriterien erfüllt sind.

Eine Änderung des individuellen Honors-Programms während dieser Zeit ist zulässig, bedarf aber der Bewilligung durch das studienrechtliche Organ.

## H.2. Abschluss

Studierende können jederzeit innerhalb der maximal erlaubten Dauer von 9 Semestern beim zuständigen studienrechtlichen Organ den Antrag auf einen *Bachelor-Abschluss with Honors* stellen. Die für einen *Bachelor-Abschluss with Honors* zu erfüllenden Kriterien sind folgende:

- Das gegenständliche reguläre Bachelor-Studium wurde mit Auszeichnung<sup>3</sup> abgeschlossen.
- Der gewichtete Gesamtnotenschnitt aller für den Abschluss des gegenständlichen Bachelor-Studiums verwendeten Lehrveranstaltungen ist  $\leq 1,5$ .
- Alle Lehrveranstaltungen des individuellen Honors-Programms wurden positiv abgeschlossen.
- Der gewichtete Gesamtnotenschnitt aller für den Abschluss des gegenständlichen Bachelor-Studiums verwendeten Lehrveranstaltungen und aller Lehrveranstaltungen des individuellen Honors-Programms ist  $\leq 1,5$ .
- Die Gesamtstudiendauer überschreitet nicht 9 Semester (gegebenenfalls unter angemessener Berücksichtigung von Beurlaubung und Teilzeit).

Als Bestätigung für den erfolgten *Bachelor-Abschluss with Honors* wird vom Rektorat der TU Wien ein Zertifikat und ein Empfehlungsschreiben ausgestellt, das die hervorragenden Studienleistungen bestätigt und die im Rahmen des individuellen Honors-Programms absolvierten zusätzlichen Lehrveranstaltungen anführt.

---

<sup>2</sup>Die für den Bachelor-Abschluss with Honors noch zu erbringenden Leistungen können in einem auf das abgeschlossene Bachelorstudium aufbauenden Masterstudium absolviert werden.

<sup>3</sup>im Sinne des Par. 73 Abs. 3 UG in der Fassung vom 26. Juni 2017