

---

# **Modulhandbuch**

Bachelor of Science (B.Sc.)

# **Computer Science**

Kohorte: Wintersemester 2020

Stand: 30. April 2020

---

---

# Inhaltsverzeichnis

---

---

|  |     |
|--|-----|
| Inhaltsverzeichnis   | 2   |
| Studiengangsbeschreibung   | 3   |
| Fachmodule der Kernqualifikation   | 6   |
| Modul M0561: Diskrete Algebraische Strukturen                            | 6   |
| Modul M0575: Prozedurale Programmierung                                  | 9   |
| Modul M0577: Nichttechnische Angebote im Bachelor                        | 13  |
| Modul M0731: Functional Programming                                      | 15  |
| Modul M0736: Linear Algebra  | 18  |
| Modul M0624: Automata Theory and Formal Languages                        | 20  |
| Modul M0737: Mathematical Analysis                                       | 24  |
| Modul M0829: Grundlagen der Betriebswirtschaftslehre                     | 26  |
| Modul M1432: Programmierparadigmen                                       | 29  |
| Modul M0730: Technische Informatik                                       | 32  |
| Modul M0834: Computernetworks and Internet Security                      | 35  |
| Modul M1423: Algorithmen und Datenstrukturen                             | 37  |
| Modul M0853: Mathematik III  | 39  |
| Modul M0562: Berechenbarkeit und Komplexität                             | 42  |
| Modul M0727: Stochastik  | 44  |
| Modul M0732: Software Engineering  | 46  |
| Modul M0852: Graphentheorie und Optimierung                              | 48  |
| Modul M0873: Software-Fachpraktikum                                      | 50  |
| Modul M1578: Seminare Informatik   | 51  |
| Modul M0672: Signale und Systeme   | 53  |
| Fachmodule der Vertiefung I. Computer- und Software-Engineering          | 57  |
| Modul M0625: Databases   | 57  |
| Modul M1586: Wissenschaftliche Programmierung                            | 60  |
| Modul M0971: Betriebssysteme   | 62  |
| Modul M0791: Rechnerarchitektur  | 64  |
| Modul M0972: Verteilte Systeme   | 66  |
| Modul M0953: Introduction to Information Security                        | 68  |
| Modul M0754: Compiler Construction                                       | 70  |
| Modul M0803: Embedded Systems  | 72  |
| Modul M1300: Software Development  | 74  |
| Fachmodule der Vertiefung II. Mathematik und Ingenieurwissenschaften     | 76  |
| Modul M0662: Numerical Mathematics I                                     | 76  |
| Modul M0854: Mathematik IV   | 79  |
| Modul M0651: Rechnergestützte Geometrie                                  | 83  |
| Modul M0941: Kombinatorische Strukturen und Algorithmen                  | 86  |
| Modul M1242: Quantenmechanik für Studierende der Ingenieurwissenschaften | 88  |
| Modul M0634: Einführung in Medizintechnische Systeme                     | 90  |
| Modul M0668: Algebraische Methoden in der Regelungstechnik               | 92  |
| Modul M1269: Labor Cyber-Physical Systems                                | 94  |
| Modul M0715: Löser für schwachbesetzte lineare Gleichungssysteme         | 96  |
| Fachmodule der Vertiefung III. Fachspezifische Fokussierung              | 98  |
| Modul M1562: Technischer Ergänzungskurs I für CSBS                       | 98  |
| Modul M1568: Technischer Ergänzungskurs II für CSBS                      | 99  |
| Thesis   | 100 |
| Modul M-001: Bachelorarbeit  | 100 |

---

---

## Studiengangsbeschreibung

---

---

### Inhalt

Die Informatik ist neben Biotechnologie, Medizintechnik und Nanotechnologie die Schlüsseltechnologie des 21. Jahrhunderts. Sie hat sich zu einer Triebfeder des technologischen Fortschrittes entwickelt, weil alle Berufszweige mit Informationsaspekten durchdrungen sind und immer neue Anwendungsfelder in der Informations- und Kommunikationstechnik erschlossen werden. Deshalb brauchen wir heute und morgen Informatikerinnen und Informatiker, die informationsverarbeitende Systeme qualifiziert und verantwortungsbewusst entwerfen, analysieren und an gegebene Einsatzbedingungen anpassen.

Die Informatik ist eng mit der Mathematik und der Elektrotechnik/Elektronik verbunden, ist aber auch als eine Basis- und Querschnittsdisziplin zu verstehen, die sich sowohl mit technischen als auch mit organisatorischen Problemen bei der Entwicklung und Anwendung informationsverarbeitender Systeme beschäftigt. Sie erforscht die grundsätzlichen Verfahrensweisen der Informationsverarbeitung und die allgemeinen Methoden der Anwendung solcher Verfahren in den verschiedensten Bereichen. Sie geht durch Abstraktion und Modellbildung sowohl über die konkreten technischen Realisierungen informationsverarbeitender Systeme als auch über die Besonderheiten spezieller Anwendungen hinaus und gelangt zur Formulierung allgemeiner Gesetzmäßigkeiten. Daraus entwickelt sie Standardlösungen für die Aufgaben der Praxis, z.B. bei der Bewältigung großer Daten- und Informationsmengen und der Steuerung komplexer Produktionsabläufe (Quelle: studienwahl.de, 05/2015).

Der Bachelorstudiengang Computer Science bietet ein wissenschaftlich fundiertes, grundlagenorientiertes Studium. Auf der Basis eines breiten und in ausgewählten Teilgebieten vertieften fachlichen Wissens werden die analytischen, kreativen und konstruktiven Fähigkeiten zur Konzipierung von informationsverarbeitenden Systemen entwickelt und gefördert. Vor allem wird die Fähigkeit zur Realisierung und Implementierung von programmierbaren Systemen erworben. Durch die Bearbeitung von vielfältigen Problemen aus verschiedenen Anwendungsbereichen entwickeln die Studierenden insgesamt eine sinnvolle Mischung aus praktischen und wissenschaftlichen Fähigkeiten. Schlüsselqualifikationen wie Teamarbeit und Präsentationstechniken werden gezielt vermittelt.

Die Informatik unterliegt schnellen Innovationen, weshalb besonderer Wert auf zukunftsfestes Wissen gelegt wird. Damit sollen die Studierenden in die Lage versetzt werden, auch die künftigen Entwicklungen der Informatik selbstständig und auf hohem Niveau in ihre berufliche Praxis und in ihren persönlichen Horizont zu integrieren. Aus diesem Grund hat der Bachelorstudiengang eine wissenschaftliche und methodenorientierte Grundausrichtung. Erworben werden vor allem gründliche Kenntnisse in Informatik und der dazu gehörigen Mathematik sowie Betriebswirtschaftslehre.

### Berufliche Perspektiven

Der Bachelorstudiengang Computer Science bereitet die Absolventen und Absolventinnen sowohl auf eine berufliche Tätigkeit im IT-Sektor als auch auf ein aufbauendes Master-Studium vor. Die Absolventen und Absolventinnen werden in die Lage versetzt, komplexe IT-Lösungen zu entwerfen und technisch umzusetzen. Ferner werden methodische Grundlagen erworben, um sich stets an neue berufliche Entwicklungen und Innovationen anzupassen. Daher sollten die Absolventen und Absolventinnen in nahezu allen Branchen eine verantwortungsvolle Tätigkeit finden können.

### Lernziele

Das Bachelorstudium Computer Science soll die Studierenden sowohl auf eine berufliche Tätigkeit als auch auf ein einschlägiges Master-Studium vorbereiten. Das hierfür notwendige methodische Grundlagenwissen wird im Rahmen des Studiums erworben. Die Lernergebnisse des Studiengangs werden durch ein Zusammenspiel von grundlegenden und weiterführenden Modulen aus Informatik, Mathematik und Betriebswirtschaftslehre erreicht. Die Lernziele sind im Folgenden eingeteilt in die Kategorien Wissen, Fertigkeiten, Sozialkompetenz und Selbstständigkeit.

### Wissen

Wissen konstituiert sich aus Fakten, Grundsätzen und Theorien und wird im Bachelorstudiengang Computer Science auf folgenden Gebieten erworben:

1. Die Absolventen und Absolventinnen kennen grundlegende Methoden und Verfahren zur mathematischen Modellbildung in der Informatik, wie etwa algebraisch spezifizierte abstrakte Datentypen, Automatenmodelle, formale Grammatiken, Modelle der Berechenbarkeit, Komplexitätsmodelle, graphentheoretische Netzwerke, Differentialgleichungen, stochastische Prozesse und dynamische Systeme im Sinne der Systemtheorie. Sie können diese beschreiben und vergleichen.
2. Die Absolventen und Absolventinnen kennen fundamentale Methoden und Verfahren zur Lösung oder Approximation von algorithmischen Entscheidungs- und Optimierungsproblemen, wie etwa automatische Differentiation, direkte erschöpfende Suche via Backtracking, Gradienten-basierte Verfahren, graphentheoretische Algorithmen, Heuristiken, lineare (ganzzahlige) Programmierung, Testen von Hypothesen, Theorem-Beweiser, sowie deren Analyse hinsichtlich Komplexität, Konvergenz und Güte. Sie sind in der Lage, diese zu skizzieren und zu diskutieren.
3. Die Absolventen und Absolventinnen kennen die Grundlagen des Software-Entwurfes und können hierbei auf gängige prozedurale, objektorientierte, funktionale und logikbasierte Programmiersprachen sowie grundlegende Datenstrukturen und Algorithmen zurückgreifen. Sie sind vertraut mit dem Betrieb von Software-Systemen unter Berücksichtigung der Organisation und Verarbeitung großer Datenmengen, der Verwaltung der zur Verfügung stehenden Betriebsmittel und einem verteilten Arrangement von Daten und Algorithmen.
4. Die Absolventen und Absolventinnen kennen den Aufbau, den Betrieb und die Organisation von Rechenanlagen und sie wissen, wie Algorithmen auf dem von-Neumann-Rechner oder einem Mikroprozessor ausgeführt werden. Sie wissen ferner, wie Hardware-Bausteine programmiertechnisch beschrieben und simuliert werden können und sie können die Einbettung eines Strukturmodells in einen technischen Rahmen skizzieren und implementieren.
5. Die Absolventen und Absolventinnen kennen eine Reihe von Anwendungsfällen valider mathematischer Modelle in der Informatik, wie etwa Algorithmen in Netzwerken, diskrete und schnelle Fourier-Transformation, Public-Key-Infrastrukturen sowie Sortier- und Suchverfahren.

### **Fertigkeiten**

Die Fähigkeit, erlerntes Wissen anzuwenden, um spezifische Probleme zu lösen, wird im Studiengang Computer Science auf vielfältige Weise unterstützt:

1. Die Absolventen und Absolventinnen sind im Stande, Instanzen formaler Modelle in der Informatik anhand einfacher Modellierungsansätze zu entwickeln, ihre Berechenbarkeit und Komplexität einzuschätzen und sie anhand geeigneter Programmierwerkzeuge umzusetzen.
2. Die Absolventen und Absolventinnen sind in der Lage, Instanzen von algorithmischen Entscheidungs- und Optimierungsproblemen unter Einsatz des Erlernten optimal oder näherungsweise zu lösen und die Lösungen zu analysieren.
3. Die Absolventen und Absolventinnen können Software-Komponenten in komplexere Softwaresysteme unter Benutzung der im Studium erarbeiteten Methoden integrieren und testen.
4. Die Absolventen und Absolventinnen werden in die Lage versetzt, Mikroprozessoren zu programmieren und Strukturbeschreibungen von einfachen Hardware-Bausteinen zu entwickeln, zu simulieren und zu bewerten.
5. Die Absolventen und Absolventinnen können vertraute Anwendungsfälle valider mathematischer Modelle aus der Informatik unter Verwendung einschlägiger Werkzeuge umsetzen und die Lösungen evaluieren.

### **Erwerb von Sozialkompetenz**

Sozialkompetenz umfasst die individuelle Fähigkeit und den Willen, zielorientiert mit anderen zusammen zu arbeiten, die Interessen der anderen zu erfassen, sich zu verständigen und die Arbeits- und Lebenswelt mitzugestalten.

1. Die Absolventen und Absolventinnen können in einem fachlich homogenen Team organisieren, spezifische Teilaufgaben übernehmen und den eigenen Beitrag reflektieren.
2. Die Absolventen und Absolventinnen sind in der Lage, sich in ein fachlich heterogenes Team einzugliedern, gemeinsame Lösungen zu erarbeiten und diese vor anderen zu vertreten.

### **Kompetenz zum selbständigen Arbeiten**

Personale Kompetenzen umfassen neben der Kompetenz zum selbständigen Handeln auch System- und Lösungskompetenzen, allgemeinen Problemstellung auf spezifische Teilprobleme abzubilden sowie die Auswahl und das Beherrschen geeigneter Methoden und Verfahren zur Problemlösung.

1. Die Absolventen und Absolventinnen können sich selbständig ein eng umrissenes Teilgebiet der Informatik erschließen und die Ergebnisse im Rahmen eines kurzen Vortrages mit fortschrittlichen Präsentationstechniken oder eines mehrseitigen Aufsatzes detailliert zusammenfassen.

2. Die Absolventen und Absolventinnen sind in der Lage, fachlich eingegrenzte Teilprojekte unter Verwendung des im Studium Erlernen in einem komplexeren IT-Entwicklungsprojekt eigenverantwortlich zu bearbeiten.

## **Studiengangsstruktur**

Das Curriculum des Bachelorstudienganges Computer Science ist wie folgt gegliedert:

- Kernqualifikation: 21 Module, 132 Leistungspunkte (LP), 1. - 6. Semester
  - Fachmodule aus dem Grundlagenbereich von Informatik und Mathematik (114 LP)
  - Software-Fachpraktikum (6 LP)
  - Grundlagen der Betriebswirtschaftslehre (6 LP)
  - Nichttechnische Ergänzungsfächer (6 LP)
- Vertiefung: 6 Module, 36 LP, 5. und 6. Semester
  - Richtungen: Computermathematik, Computer- und Software-Engineering
- Bachelorarbeit: 12 LP, 6. Semester

Damit ergibt sich ein Gesamtaufwand von 180 LP.

Der Studienplan ist mit einem Mobilitätsfenster versehen, so dass das fünfte Semester unter Umständen im Ausland absolviert werden kann.

## Fachmodule der Kernqualifikation

Im Studium der Informatik werden zunächst die Grundlagen der praktischen, theoretischen und technischen Informatik sowie der dazu benötigten Mathematik gelehrt. An der TUHH wird großer Wert auf eine breite methodenorientierte Ausbildung gelegt.

Im Einzelnen werden folgende Lehrinhalte vermittelt:

- Praktische Informatik
  - Programmier-Paradigmen (funktional, imperativ, objektorientiert)
  - Algorithmentechnik
  - Datenstrukturen
  - Softwareentwicklung
- Theoretische Informatik
  - Automatentheorie und Formale Sprachen
  - Berechenbarkeit und Komplexitätstheorie
  - Graphentheorie und Optimierung
  - Informationssicherheit
- Technische Informatik
  - Digitaltechnik
  - Rechnerstrukturen
  - Betriebssysteme
  - Rechnernetze
- Mathematik
  - Diskrete Mathematik
  - Lineare Algebra
  - Analysis
  - Stochastik
  - Klassische Systemtheorie

Abgerundet wird das Kernangebot durch Veranstaltungen, welche die persönlichen, sozialen und methodischen Kompetenzen (Soft-Skills) erhöhen sollen:

- Einführung in die Betriebswirtschaftslehre
- Studium Generale (3 Veranstaltungen frei wählbar aus dem Angebot)
- Seminare Informatik und Mathematik (3 Seminare frei wählbar aus dem Angebot)
- Software-Fachpraktikum (sechswöchig, wird in der Regel in der Industrie absolviert)

### Modul M0561: Diskrete Algebraische Strukturen

#### Lehrveranstaltungen

| Titel                                    | Typ          | SWS | LP |
|--|--------------|-----|----|
| Diskrete Algebraische Strukturen (L0164) | Vorlesung    | 2   | 3  |
| Diskrete Algebraische Strukturen (L0165) | Gruppenübung | 2   | 3  |

**Modulverantwortlicher** Prof. Karl-Heinz Zimmermann

**Zulassungsvoraussetzungen** Keine

**Empfohlene Vorkenntnisse** Abiturkenntnisse in Mathematik.

**Modulziele/ angestrebte Lernergebnisse** Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht

|                      |   |
|----------------------|---|
| <b>Fachkompetenz</b> | Wissen: Die Studierenden kennen <ul style="list-style-type: none"> <li>• zahlentheoretische und funktionsbasierte Modelle der Kryptographie sowie Grundlagen der linearen Codes;</li> <li>• den Aufbau und Struktur von Restklassenringen (Euklidische Ringe) und endlichen Körpern;</li> <li>• den Aufbau und die Struktur von Unter-, Summen- und Faktorstrukturen in algebraischen Gebilden sowie Homomorphismen zwischen diesen Strukturen;</li> <li>• den Aufbau und die Abzählung von elementaren kombinatorischen Strukturen;</li> </ul> |
| <i>Wissen</i>        |   |

|   |  |
|---|--|
|   | <ul style="list-style-type: none"> <li>• die wichtigsten Beweiskonzepte der modernen Mathematik;</li> <li>• den Aufbau der höheren Mathematik basierend auf mathematischer Logik und Mengenlehre;</li> <li>• grundlegende Aspekte des Einsatzes von mathematischer Software (Computeralgebrasystem Maple) zur Lösung von algebraischen oder kombinatorischen Aufgabenstellungen.</li> </ul>  |
| <i>Fertigkeiten</i>                     | <p>Fertigkeiten: Die Studierenden können</p> <ul style="list-style-type: none"> <li>• in Restklassenringen (Euklidischen Ringen) rechnen;</li> <li>• Unter-, Summen- und Faktorstrukturen in algebraischen Gebilden aufstellen und in ihnen rechnen sowie algebraische Strukturen durch Homomorphismen aufeinander beziehen;</li> <li>• elementar-kombinatorische Strukturen identifizieren und abzählen;</li> <li>• die Sprache der Mathematik, basierend auf Mathematischer Logik und Mengenlehre, dienstbar verwenden;</li> <li>• einfache, im Kontext stehende mathematische Aussagen beweisen;</li> <li>• einschlägige mathematische Software (Computeralgebrasystem Maple) zielgerichtet einsetzen.</li> </ul> |
| <b>Personale Kompetenzen</b>            |  |
| <i>Sozialkompetenz</i>                  | Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.   |
| <i>Selbstständigkeit</i>                | Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachbüchern selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.  |
| <b>Arbeitsaufwand in Stunden</b>        | Eigenstudium 124, Präsenzstudium 56  |
| <b>Leistungspunkte</b>                  | 6  |
| <b>Studienleistung</b>                  | Keine  |
| <b>Prüfung</b>                          | Klausur  |
| <b>Prüfungsdauer und -umfang</b>        | 120 min  |
| <b>Zuordnung zu folgenden Curricula</b> | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Orientierungsstudium: Kernqualifikation: Wahlpflicht   |

| Lehrveranstaltung L0164: Diskrete Algebraische Strukturen |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Vorlesung                          |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                          | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | WiSe                               |
| <b>Inhalt</b>   |                                    |
| <b>Literatur</b>  |                                    |

| <b>Lehrveranstaltung L0165: Diskrete Algebraische Strukturen</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>  | DE/EN                              |
| <b>Zeitraum</b>  | WiSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

## Modul M0575: Prozedurale Programmierung

### Lehrveranstaltungen

| Titel                              | Typ            | SWS | LP |
|------------------------------------|----------------|-----|----|
| Prozedurale Programmierung (L0197) | Vorlesung      | 1   | 2  |
| Prozedurale Programmierung (L0201) | Hörsaalübung   | 1   | 1  |
| Prozedurale Programmierung (L0202) | Laborpraktikum | 2   | 3  |

|   |   |
|---|---|
| <b>Modulverantwortlicher</b>                  | Prof. Siegfried Rump  |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |
| <b>Empfohlene Vorkenntnisse</b>               | Elementare Handhabung eines PC<br>Elementare Mathematikkenntnisse   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |
| <b>Fachkompetenz</b>                          | <p>Die Studierenden erwerben folgendes Wissen:</p> <ul style="list-style-type: none"> <li>• Sie kennen elementare Sprachelemente der Programmiersprache C. Sie kennen die grundlegenden Datentypen und wissen um ihre Einsatzgebiete.</li> <li>• Sie haben ein Verständnis davon, was die Aufgaben eines Compilers, des Präprozessors und der Entwicklungsumgebung sind und wie diese interagieren.</li> <li>• Sie beherrschen die Einbindung und Verwendung externer Programm-Bibliotheken zur Erweiterung des Funktionsumfangs.</li> <li>• Sie wissen, wie man Header-Dateien verwendet und Funktionsschnittstellen festlegt, um größere Programmierprojekte kreieren zu können.</li> <li>• Sie haben ein Verständnis dafür, wie das implementierte Programm mit dem Betriebssystem interagiert. Dies befähigt Sie dazu, Programme zu entwickeln, welche Eingaben des Benutzers, Betriebseingaben oder auch entsprechende Dateien verarbeiten und gewünschte Ausgaben erzeugen.</li> <li>• Sie haben mehrere Herangehensweisen zur Implementierung häufig verwendeter Algorithmen gelernt.</li> </ul> <p><i>Wissen</i></p> <ul style="list-style-type: none"> <li>• Die Studierenden sind in der Lage, die Komplexität eines Algorithmus zu bewerten und eine effiziente Implementierung vorzunehmen.</li> <li>• Die Studierenden können Algorithmen für eine Vielzahl von Funktionalitäten modellieren und programmieren. Zudem können Sie die Implementierung an eine vorgegebene API anpassen.</li> </ul> <p><i>Fertigkeiten</i></p> |
| <b>Personale Kompetenzen</b>                  | <p>Die Studierenden erwerben folgende Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Sie können in Kleingruppen Aufgaben gemeinsam lösen, Programmfehler analysieren und beheben und ihr</li> </ul>  |

|   |   |
|---|---|
| <i>Sozialkompetenz</i>                  | <p>erzieltes Ergebnis gemeinsam präsentieren.</p> <ul style="list-style-type: none"> <li>• Sie können sich Sachverhalte direkt am Rechner durch einfaches Ausprobieren gegenseitig klar machen.</li> <li>• Sie können in Kleingruppen gemeinsam eine Projektidee und -planung erarbeiten.</li> <li>• Sie müssen den betreuenden Tutoren ihre eigenen Lösungsansätze verständlich kommunizieren und ihre Programme präsentieren.</li> </ul>  |
| <i>Selbstständigkeit</i>                | <ul style="list-style-type: none"> <li>• Die Studierenden müssen in Einzeltestaten sowie einer abschließenden Prüfung ihre Programmierfertigkeiten unter Beweis stellen und selbständig ihr erlerntes Wissen zur Lösung neuer Aufgabenstellungen anwenden.</li> <li>• Die Studierenden haben die Möglichkeit, ihre erlernten Fähigkeiten beim Lösen einer Vielzahl von Präsenzaufgaben zu überprüfen.</li> <li>• Zur effizienten Bearbeitung der Aufgaben des Praktikums teilen die Studierenden innerhalb ihrer Gruppen die Übungsaufgaben auf. Jeder Studierende muss zunächst selbständig eine Teilaufgabe lösen.</li> </ul> |
| <b>Arbeitsaufwand in Stunden</b>        | Eigenstudium 124, Präsenzstudium 56   |
| <b>Leistungspunkte</b>                  | 6   |
| <b>Studienleistung</b>                  | Keine   |
| <b>Prüfung</b>                          | Klausur   |
| <b>Prüfungsdauer und -umfang</b>        | 90 Minuten  |
| <b>Zuordnung zu folgenden Curricula</b> | <p>Computer Science: Kernqualifikation: Pflicht<br/>                 Data Science: Kernqualifikation: Pflicht<br/>                 Elektrotechnik: Kernqualifikation: Pflicht<br/>                 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br/>                 Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht<br/>                 Mechatronik: Kernqualifikation: Pflicht<br/>                 Orientierungsstudium: Kernqualifikation: Wahlpflicht<br/>                 Technomathematik: Kernqualifikation: Pflicht</p>  |

| Lehrveranstaltung L0197: Prozedurale Programmierung |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 1  |
| <b>LP</b>   | 2  |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 46, Präsenzstudium 14   |
| <b>Dozenten</b>                                     | Prof. Siegfried Rump   |
| <b>Sprachen</b>                                     | DE   |
| <b>Zeitraum</b>                                     | WiSe   |
| <b>Inhalt</b>                                       | <ul style="list-style-type: none"> <li>• elementare Datentypen (Integer, Gleitpunktformat, ASCII-Zeichen) und ihre Abhängigkeiten von der Architektur</li> <li>• höhere Datentypen (Zeiger, Arrays, Strings, Strukturen, Listen)</li> <li>• Operatoren (arithmetische Operationen, logische Operationen, Bit-Operationen)</li> <li>• Kontrollflussstrukturen (bedingte Verzweigung, Schleifen, Sprünge)</li> <li>• Präprozessor-Direktiven (Makros, bedingte Kompilierung, modulares Design)</li> <li>• Funktionen (Funktionsdefinition/-interface, Rekursion, "call by value" versus "call by reference", Funktionszeiger)</li> <li>• essentielle Standard-Bibliotheken und -Funktionen (stdio.h, stdlib.h, math.h, string.h, time.h)</li> <li>• Dateikonzept, Streams</li> <li>• einfache Algorithmen (Sortierfunktionen, Reihenentwicklung, gleichverteilte Permutation)</li> <li>• Übungsprogramme zur Vertiefung der Programmierkenntnisse</li> </ul> |
| <b>Literatur</b>                                    | <p><b>Kernighan, Brian W</b> (Ritchie, Dennis M.)<br/>                     The C programming language<br/>                     ISBN: 9780131103702<br/> <i>Upper Saddle River, NJ [u.a.] : Prentice Hall PTR, 2009</i></p> <p><b>Sedgewick, Robert</b><br/>                     Algorithms in C<br/>                     ISBN: 0201316633<br/> <i>Reading, Mass. [u.a.] : Addison-Wesley, 2007</i></p> <p><b>Kaiser, Ulrich</b> (Kecher, Christoph.)<br/>                     C/C++: Von den Grundlagen zur professionellen Programmierung<br/>                     ISBN: 9783898428392<br/> <i>Bonn : Galileo Press, 2010</i></p> <p><b>Wolf, Jürgen</b><br/>                     C von A bis Z : das umfassende Handbuch<br/>                     ISBN: 3836214113<br/> <i>Bonn : Galileo Press, 2009</i></p>  |

| Lehrveranstaltung L0201: Prozedurale Programmierung |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Hörsaalübung                       |
| <b>SWS</b>  | 1                                  |
| <b>LP</b>   | 1                                  |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 16, Präsenzstudium 14 |
| <b>Dozenten</b>                                     | Prof. Siegfried Rump               |
| <b>Sprachen</b>                                     | DE                                 |
| <b>Zeitraum</b>                                     | WiSe                               |
| <b>Inhalt</b>                                       | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                    | Siehe korrespondierende Vorlesung  |

| <b>Lehrveranstaltung L0202: Prozedurale Programmierung</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Laborpraktikum                     |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                           | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Siegfried Rump               |
| <b>Sprachen</b>  | DE                                 |
| <b>Zeitraum</b>  | WiSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

## Modul M0577: Nichttechnische Angebote im Bachelor

|   |   |
|---|---|
| <b>Modulverantwortlicher</b>                  | Dagmar Richter  |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |
| <b>Empfohlene Vorkenntnisse</b>               | Keine   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |
| <b>Fachkompetenz</b>                          | <p><b>Die Nichttechnischen Angebote (NTA)</b></p> <p>vermitteln die in Hinblick auf das Ausbildungsprofil der TUHH nötigen Kompetenzen, die ingenieurwissenschaftliche Fachlehre fördern aber nicht abschließend behandeln kann: Eigenverantwortlichkeit, Selbstführung, Zusammenarbeit und fachliche wie personale Leitungsbefähigung der zukünftigen Ingenieurinnen und Ingenieure. Er setzt diese Ausbildungsziele in seiner <b>Lehrarchitektur</b>, den <b>Lehr-Lern-Arrangements</b>, den <b>Lehrbereichen</b> und durch Lehrangebote um, in denen sich Studierende wahlweise für <b>spezifische Kompetenzen</b> und ein <b>Kompetenzniveau</b> auf Bachelor- oder Masterebene qualifizieren können. Die Lehrangebote sind jeweils in einem Modulkatalog Nichttechnische Ergänzungskurse zusammengefasst.</p> <p><b>Die Lehrarchitektur</b></p> <p>besteht aus einem studiengangübergreifenden Pflichtstudienangebot. Durch dieses zentral konzipierte Lehrangebot wird die Profilierung der TUHH Ausbildung auch im Nichttechnischen Bereich gewährleistet.</p> <p>Die Lernarchitektur erfordert und übt eigenverantwortliche Bildungsplanung in Hinblick auf den individuellen Kompetenzaufbau ein und stellt dazu Orientierungswissen zu thematischen Schwerpunkten von Veranstaltungen bereit.</p> <p>Das über den gesamten Studienverlauf begleitend studierbare Angebot kann ggf. in ein-zwei Semestern studiert werden. Angesichts der bekannten, individuellen Anpassungsprobleme beim Übergang von Schule zu Hochschule in den ersten Semestern und um individuell geplante Auslandsemester zu fördern, wird jedoch von einer Studienfixierung in konkreten Fachsemestern abgesehen.</p> <p><b>Die Lehr-Lern-Arrangements</b></p> <p>sehen für Studierende - nach B.Sc. und M.Sc. getrennt - ein semester- und fachübergreifendes voneinander Lernen vor. Der Umgang mit Interdisziplinarität und einer Vielfalt von Lernständen in Veranstaltungen wird eingeübt - und in spezifischen Veranstaltungen gezielt gefördert.</p> <p><b>Die Lehrbereiche</b></p> <p>basieren auf Forschungsergebnissen aus den wissenschaftlichen Disziplinen Kulturwissenschaften, Gesellschaftswissenschaften, Kunst, <i>Wissen</i> Geschichtswissenschaften, Kommunikationswissenschaften, Migrationswissenschaften, Nachhaltigkeitsforschung und aus der Fachdidaktik der Ingenieurwissenschaften. Über alle Studiengänge hinweg besteht im Bachelorbereich zusätzlich ab Wintersemester 2014/15 das Angebot, gezielt Betriebswirtschaftliches und Gründungswissen aufzubauen. Das Lehrangebot wird durch soft skill und Fremdsprachkurse ergänzt. Hier werden insbesondere kommunikative Kompetenzen z.B. für Outgoing Engineers gezielt gefördert.</p> <p><b>Das Kompetenzniveau</b></p> <p>der Veranstaltungen in den Modulen der nichttechnischen Ergänzungskurse unterscheidet sich in Hinblick auf das zugrunde gelegte Ausbildungsziel: Diese Unterschiede spiegeln sich in den verwendeten Praxisbeispielen, in den - auf unterschiedliche berufliche Anwendungskontexte verweisende - Inhalten und im für M.Sc. stärker wissenschaftlich-theoretischen Abstraktionsniveau. Die Soft skills für Bachelor- und für Masterabsolventinnen/ Absolventen unterscheidet sich an Hand der im Berufsleben unterschiedlichen Positionen im Team und bei der Anleitung von Gruppen.</p> <p><b>Fachkompetenz (Wissen)</b></p> <p>Die Studierenden können</p> <ul style="list-style-type: none"> <li>• ausgewählte Spezialgebiete innerhalb der jeweiligen nichttechnischen</li> </ul> |

|  |   |
|--|---|
| <p><i>Fertigkeiten</i></p> <p><b>Personale Kompetenzen</b></p> <p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i></p> | <p>Mutterdisziplinen verorten,</p> <ul style="list-style-type: none"> <li>• in den im Lehrbereich vertretenen Disziplinen grundlegende Theorien, Kategorien, Begrifflichkeiten, Modelle, Konzepte oder künstlerischen Techniken skizzieren,</li> <li>• diese fremden Fachdisziplinen systematisch auf die eigene Disziplin beziehen, d.h. sowohl abgrenzen als auch Anschlüsse benennen,</li> <li>• in Grundzügen skizzieren, inwiefern wissenschaftliche Disziplinen, Paradigmen, Modelle, Instrumente, Verfahrensweisen und Repräsentationsformen der Fachwissenschaften einer individuellen und soziokulturellen Interpretation und Historizität unterliegen,</li> <li>• können Gegenstandsangemessen in einer Fremdsprache kommunizieren (sofern dies der gewählte Schwerpunkt im nichttechnischen Bereich ist).</li> </ul> <p>Die Studierenden können in ausgewählten Teilbereichen</p> <ul style="list-style-type: none"> <li>• grundlegende Methoden der genannten Wissenschaftsdisziplinen anwenden.</li> <li>• technische Phänomene, Modelle, Theorien usw. aus der Perspektive einer anderen, oben erwähnten Fachdisziplin befragen.</li> <li>• einfache Problemstellungen aus den behandelten Wissenschaftsdisziplinen erfolgreich bearbeiten,</li> <li>• bei praktischen Fragestellungen in Kontexten, die den technischen Sach- und Fachbezug übersteigen, ihre Entscheidungen zu Organisations- und Anwendungsformen der Technik begründen.</li> </ul> <p>Die Studierenden sind fähig ,</p> <ul style="list-style-type: none"> <li>• in unterschiedlichem Ausmaß kooperativ zu lernen</li> <li>• eigene Aufgabenstellungen in den o.g. Bereichen in adressatengerechter Weise in einer Partner- oder Gruppensituation zu präsentieren und zu analysieren,</li> <li>• nichttechnische Fragestellungen einer Zuhörerschaft mit technischem Hintergrund verständlich darzustellen</li> <li>• sich landessprachlich kompetent, kulturell angemessen und geschlechtersensibel auszudrücken (sofern dies der gewählte Schwerpunkt im NTW-Bereich ist) .</li> </ul> <p>Die Studierenden sind in ausgewählten Bereichen in der Lage,</p> <ul style="list-style-type: none"> <li>• die eigene Profession und Professionalität im Kontext der lebensweltlichen Anwendungsgebiete zu reflektieren,</li> <li>• sich selbst und die eigenen Lernprozesse zu organisieren,</li> <li>• Fragestellungen vor einem breiten Bildungshorizont zu reflektieren und verantwortlich zu entscheiden,</li> <li>• sich in Bezug auf ein nichttechnisches Sachthema mündlich oder schriftlich kompetent auszudrücken.</li> <li>• sich als unternehmerisches Subjekt zu organisieren, (sofern dies ein gewählter Schwerpunkt im NTW-Bereich ist).</li> </ul> |
| <p><b>Arbeitsaufwand in Stunden</b></p>  | <p>Abhängig von der Wahl der Lehrveranstaltungen</p>  |
| <p><b>Leistungspunkte</b></p>  | <p>6</p>  |

**Lehrveranstaltungen**

**Die Informationen zu den Lehrveranstaltungen entnehmen Sie dem separat veröffentlichten Modulhandbuch des Moduls.**

| Modul M0731: Functional Programming           |  |               |  |
|---|--|---------------|--|
| <b>Lehrveranstaltungen</b>                    |  |               |  |
| <b>Titel</b>                                  |  | <b>Typ</b>    | <b>SWS</b> <b>LP</b>                               |
| Funktionales Programmieren (L0624)            |  | Vorlesung     | 2            2                                     |
| Funktionales Programmieren (L0625)            |  | Hörsaalübung  | 2            2                                     |
| Funktionales Programmieren (L0626)            |  | Gruppenübung  | 2            2                                     |
| <b>Modulverantwortlicher</b>                  | Prof. Sibylle Schupp   |               |  |
| <b>Zulassungsvoraussetzungen</b>              | None   |               |  |
| <b>Empfohlene Vorkenntnisse</b>               | Discrete mathematics at high-school level  |               |  |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |               |  |
| <b>Fachkompetenz</b>                          |  |               |  |
| <i>Wissen</i>                                 | Students apply the principles, constructs, and simple design techniques of functional programming. They demonstrate their ability to read Haskell programs and to explain Haskell syntax as well as Haskell's read-eval-print loop. They interpret warnings and find errors in programs. They apply the fundamental data structures, data types, and type constructors. They employ strategies for unit tests of functions and simple proof techniques for partial and total correctness. They distinguish laziness from other evaluation strategies.  |               |  |
| <i>Fertigkeiten</i>                           | Students break a natural-language description down in parts amenable to a formal specification and develop a functional program in a structured way. They assess different language constructs, make conscious selections both at specification and implementations level, and justify their choice. They analyze given programs and rewrite them in a controlled way. They design and implement unit tests and can assess the quality of their tests. They argue for the correctness of their program.  |               |  |
| <b>Personale Kompetenzen</b>                  |  |               |  |
| <i>Sozialkompetenz</i>                        | Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.  |               |  |
| <i>Selbstständigkeit</i>                      | In programming labs, students learn under supervision (a.k.a. "Betreutes Programmieren") the mechanics of programming. In exercises, they develop solutions individually and independently, and receive feedback.  |               |  |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 96, Präsenzstudium 84   |               |  |
| <b>Leistungspunkte</b>                        | 6  |               |  |
| <b>Studienleistung</b>                        | <b>Verpflichtend</b>   | <b>Anteil</b> | <b>Art der Studienleistung</b> <b>Beschreibung</b> |
|   | Ja   | 15 %          | Übungsaufgaben                                     |
| <b>Prüfung</b>                                | Klausur  |               |  |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |               |  |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Wahlpflicht<br>Data Science: Technischer Ergänzungskurs: Wahlpflicht<br>Engineering Science: Vertiefung Mechatronics: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht |               |  |

| <b>Lehrveranstaltung L0624: Functional Programming</b> |   |
|--|---|
| <b>Typ</b>   | Vorlesung   |
| <b>SWS</b>   | 2   |
| <b>LP</b>  | 2   |
| <b>Arbeitsaufwand in Stunden</b>                       | Eigenstudium 32, Präsenzstudium 28  |
| <b>Dozenten</b>  | Prof. Sibylle Schupp  |
| <b>Sprachen</b>  | EN  |
| <b>Zeitraum</b>  | WiSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul> |
| <b>Literatur</b>                                       | Graham Hutton, Programming in Haskell, Cambridge University Press 2007.   |

| <b>Lehrveranstaltung L0625: Functional Programming</b> |   |
|--|---|
| <b>Typ</b>   | Hörsaalübung  |
| <b>SWS</b>   | 2   |
| <b>LP</b>  | 2   |
| <b>Arbeitsaufwand in Stunden</b>                       | Eigenstudium 32, Präsenzstudium 28  |
| <b>Dozenten</b>  | Prof. Sibylle Schupp  |
| <b>Sprachen</b>  | EN  |
| <b>Zeitraum</b>  | WiSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul> |
| <b>Literatur</b>                                       | Graham Hutton, Programming in Haskell, Cambridge University Press 2007.   |

| <b>Lehrveranstaltung L0626: Functional Programming</b> |   |
|--|---|
| <b>Typ</b>   | Gruppenübung  |
| <b>SWS</b>   | 2   |
| <b>LP</b>  | 2   |
| <b>Arbeitsaufwand in Stunden</b>                       | Eigenstudium 32, Präsenzstudium 28  |
| <b>Dozenten</b>  | Prof. Sibylle Schupp  |
| <b>Sprachen</b>  | EN  |
| <b>Zeitraum</b>  | WiSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul> |
| <b>Literatur</b>                                       | Graham Hutton, Programming in Haskell, Cambridge University Press 2007.   |

| Modul M0736: Linear Algebra                   |  |              |            |           |
|---|--|--------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |              |            |           |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Lineare Algebra (L0642)                       |  | Vorlesung    | 4          | 4         |
| Lineare Algebra (L0643)                       |  | Hörsaalübung | 2          | 2         |
| Lineare Algebra (L0645)                       |  | Gruppenübung | 2          | 2         |
| <b>Modulverantwortlicher</b>                  | Prof. Daniel Ruprecht  |              |            |           |
| <b>Zulassungsvoraussetzungen</b>              | None   |              |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | None   |              |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |            |           |
| <b>Fachkompetenz</b>                          |  |              |            |           |
| <i>Wissen</i>                                 | <ul style="list-style-type: none"> <li>• Students can name the basic concepts in linear algebra. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul>  |              |            |           |
| <i>Fertigkeiten</i>                           | <ul style="list-style-type: none"> <li>• Students can model problems in linear algebra with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> |              |            |           |
| <b>Personale Kompetenzen</b>                  |  |              |            |           |
| <i>Sozialkompetenz</i>                        | - Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) and to present their results appropriately (e.g. during exercise class).   |              |            |           |
| <i>Selbstständigkeit</i>                      | - Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.<br>- Students can put their knowledge in relation to the contents of other lectures.<br>- Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.  |              |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 128, Präsenzstudium 112   |              |            |           |
| <b>Leistungspunkte</b>                        | 8  |              |            |           |
| <b>Studienleistung</b>                        | Keine  |              |            |           |
| <b>Prüfung</b>                                | Klausur  |              |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 120  |              |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Engineering Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Kernqualifikation: Pflicht  |              |            |           |

| <b>Lehrveranstaltung L0642: Linear Algebra</b> |   |
|--|---|
| <b>Typ</b>                                     | Vorlesung   |
| <b>SWS</b>                                     | 4   |
| <b>LP</b>                                      | 4   |
| <b>Arbeitsaufwand in Stunden</b>               | Eigenstudium 64, Präsenzstudium 56  |
| <b>Dozenten</b>                                | Dr. Julian Großmann   |
| <b>Sprachen</b>                                | EN  |
| <b>Zeitraum</b>                                | WiSe  |
| <b>Inhalt</b>                                  | Preliminaries<br>Vector spaces<br>Matrices and linear systems of equations<br>Scalar products and orthogonality<br>Basis transformation<br>Determinants<br>Eigen values |
| <b>Literatur</b>                               | Strang: Linear Algebra<br>Beutelsbacher: Lineare Algebra  |

| <b>Lehrveranstaltung L0643: Linear Algebra</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>                                     | Hörsaalübung                       |
| <b>SWS</b>                                     | 2                                  |
| <b>LP</b>                                      | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>               | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                                | Dr. Julian Großmann, Jan Meichsner |
| <b>Sprachen</b>                                | EN                                 |
| <b>Zeitraum</b>                                | WiSe                               |
| <b>Inhalt</b>                                  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                               | Siehe korrespondierende Vorlesung  |

| <b>Lehrveranstaltung L0645: Linear Algebra</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>                                     | Gruppenübung                       |
| <b>SWS</b>                                     | 2                                  |
| <b>LP</b>                                      | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>               | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                                | Dr. Julian Großmann                |
| <b>Sprachen</b>                                | EN                                 |
| <b>Zeitraum</b>                                | WiSe                               |
| <b>Inhalt</b>                                  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                               | Siehe korrespondierende Vorlesung  |

| <b>Modul M0624: Automata Theory and Formal Languages</b> |  |
|--|--|
| <b>Lehrveranstaltungen</b>                               |  |
| <b>Titel</b>   | <b>Typ</b> <b>SWS</b> <b>LP</b>  |
| Automatentheorie und Formale Sprachen (L0332)            | Vorlesung                      2                      4  |
| Automatentheorie und Formale Sprachen (L0507)            | Gruppenübung                      2                      2   |
| <b>Modulverantwortlicher</b>                             | Prof. Tobias Knopp   |
| <b>Zulassungsvoraussetzungen</b>                         | None   |
| <b>Empfohlene Vorkenntnisse</b>                          | <p>Participating students should be able to</p> <ul style="list-style-type: none"> <li>- specify algorithms for simple data structures (such as, e.g., arrays) to solve computational problems</li> <li>- apply propositional logic and predicate logic for specifying and understanding mathematical proofs</li> <li>- apply the knowledge and skills taught in the module Discrete Algebraic Structures</li> </ul>   |
| <b>Modulziele/ angestrebte Lernergebnisse</b>            | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                                     | <p>Students can explain syntax, semantics, and decision problems of propositional logic, and they are able to give algorithms for solving decision problems. Students can show correspondences to Boolean algebra. Students can describe which application problems are hard to represent with propositional logic, and therefore, the students can motivate predicate logic, and define syntax, semantics, and decision problems for this representation formalism. Students can explain unification and resolution for solving the predicate logic SAT decision problem. Students can also describe syntax, semantics, and decision problems for various kinds of temporal logic, and identify their application areas. The participants of the course can define various kinds of finite automata and can identify relationships to logic and formal grammars. The spectrum that students can explain ranges from deterministic and nondeterministic finite automata and pushdown automata to Turing machines. Students can name those formalism for which nondeterminism is more expressive than determinism. They are also able to demonstrate which decision problems require which expressivity, and, in addition, students can transform decision problems w.r.t. one formalism into decision problems w.r.t. other formalisms. They understand that some formalisms easily induce algorithms whereas others are best suited for specifying systems and their properties. Students can describe the relationships between formalisms such as logic, automata, or grammars.</p> |
| <i>Wissen</i>  |  |
| <i>Fertigkeiten</i>                                      | <p>Students can apply propositional logic as well as predicate logic resolution to a given set of formulas. Students analyze application problems in order to derive propositional logic, predicate logic, or temporal logic formulas to represent them. They can evaluate which formalism is best suited for a particular application problem, and they can demonstrate the application of algorithms for decision problems to specific formulas. Students can also transform nondeterministic automata into deterministic ones, or derive grammars from automata and vice versa. They can show how parsers work, and they can apply algorithms for the language emptiness problem in case of infinite words.</p>   |
| <b>Personale Kompetenzen</b>                             |  |
| <i>Sozialkompetenz</i>                                   |  |
| <i>Selbstständigkeit</i>                                 |  |
| <b>Arbeitsaufwand in Stunden</b>                         | Eigenstudium 124, Präsenzstudium 56  |
| <b>Leistungspunkte</b>                                   | 6  |
| <b>Studienleistung</b>                                   | Keine  |
| <b>Prüfung</b>   | Klausur  |
| <b>Prüfungsdauer und -umfang</b>                         | 90 min   |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Computer Science: Kernqualifikation: Pflicht   |

|   |  |
|---|--|
| <b>Zuordnung zu folgenden Curricula</b> | Data Science: Kernqualifikation: Pflicht<br>Engineering Science: Vertiefung Mechatronics: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Orientierungsstudium: Kernqualifikation: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht |
|---|--|

| Lehrveranstaltung L0332: Automata Theory and Formal Languages |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 4   |
| <b>Arbeitsaufwand in Stunden</b>                              | Eigenstudium 92, Präsenzstudium 28  |
| <b>Dozenten</b>   | Prof. Tobias Knopp  |
| <b>Sprachen</b>   | EN  |
| <b>Zeitraum</b>   | SoSe  |
| <b>Inhalt</b>   | <ol style="list-style-type: none"> <li>1. Propositional logic, Boolean algebra, propositional resolution, SAT-2KNF</li> <li>2. Predicate logic, unification, predicate logic resolution</li> <li>3. Temporal Logics (LTL, CTL)</li> <li>4. Deterministic finite automata, definition and construction</li> <li>5. Regular languages, closure properties, word problem, string matching</li> <li>6. Nondeterministic automata:<br/>Rabin-Scott transformation of nondeterministic into deterministic automata</li> <li>7. Epsilon automata, minimization of automata, elimination of e-edges, uniqueness of the minimal automaton (modulo renaming of states)</li> <li>8. Myhill-Nerode Theorem:<br/>Correctness of the minimization procedure, equivalence classes of strings induced by automata</li> <li>9. Pumping Lemma for regular languages:<br/>provision of a tool which, in some cases, can be used to show that a finite automaton principally cannot be expressive enough to solve a word problem for some given language</li> <li>10. Regular expressions vs. finite automata:<br/>Equivalence of formalisms, systematic transformation of representations, reductions</li> <li>11. Pushdown automata and context-free grammars:<br/>Definition of pushdown automata, definition of context-free grammars, derivations, parse trees, ambiguities, pumping lemma for context-free grammars, transformation of formalisms (from pushdown automata to context-free grammars and back)</li> <li>12. Chomsky normal form</li> <li>13. CYK algorithm for deciding the word problem for context-free grammars</li> <li>14. Deterministic pushdown automata</li> <li>15. Deterministic vs. nondeterministic pushdown automata:<br/>Application for parsing, LL(k) or LR(k) grammars and parsers vs. deterministic pushdown automata, compiler compiler</li> <li>16. Regular grammars</li> <li>17. Outlook: Turing machines and linear bounded automata vs general and context-sensitive grammars</li> <li>18. Chomsky hierarchy</li> <li>19. Mealy- and Moore automata:<br/>Automata with output (w/o accepting states), infinite state sequences, automata networks</li> <li>20. Omega automata: Automata for infinite input words, Büchi automata, representation of state transition systems, verification w.r.t. temporal logic specifications (in particular LTL)</li> <li>21. LTL safety conditions and model checking with Büchi automata, relationships between automata and logic</li> <li>22. Fixed points, propositional mu-calculus</li> <li>23. Characterization of regular languages by monadic second-order logic (MSO)</li> </ol> |
| <b>Literatur</b>  | <ol style="list-style-type: none"> <li>1. Logik für Informatiker Uwe Schöning, Spektrum, 5. Aufl.</li> <li>2. Logik für Informatiker Martin Kreuzer, Stefan Kühling, Pearson Studium, 2006</li> <li>3. Grundkurs Theoretische Informatik, Gottfried Vossen, Kurt-Ulrich Witt, Vieweg-Verlag, 2010.</li> <li>4. Principles of Model Checking, Christel Baier, Joost-Pieter Katoen, The MIT Press, 2007</li> </ol>  |

| <b>Lehrveranstaltung L0507: Automata Theory and Formal Languages</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                                     | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Tobias Knopp                 |
| <b>Sprachen</b>  | EN                                 |
| <b>Zeitraum</b>  | SoSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

| Modul M0737: Mathematical Analysis            |   |  |            |           |
|---|---|--|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |   |  |            |           |
| <b>Titel</b>                                  |   | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Mathematische Analysis (L0647)                |   | Vorlesung  | 4          | 4         |
| Mathematische Analysis (L0648)                |   | Hörsaalübung   | 2          | 2         |
| Mathematische Analysis (L0649)                |   | Gruppenübung   | 2          | 2         |
| <b>Modulverantwortlicher</b>                  | Prof. Daniel Ruprecht   |  |            |           |
| <b>Zulassungsvoraussetzungen</b>              | None  |  |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | None  |  |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |  |            |           |
| <b>Fachkompetenz</b>                          |   |  |            |           |
| <i>Wissen</i>                                 |   | <ul style="list-style-type: none"> <li>• Students can name the basic concepts in analysis. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul>  |            |           |
| <i>Fertigkeiten</i>                           |   | <ul style="list-style-type: none"> <li>• Students can model problems in analysis with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> |            |           |
| <b>Personale Kompetenzen</b>                  |   |  |            |           |
| <i>Sozialkompetenz</i>                        |   | - Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) and to present their results appropriately (e.g. during exercise class).   |            |           |
| <i>Selbstständigkeit</i>                      |   | - Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.<br>- Students can put their knowledge in relation to the contents of other lectures.<br>- Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.  |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 128, Präsenzstudium 112  |  |            |           |
| <b>Leistungspunkte</b>                        | 8   |  |            |           |
| <b>Studienleistung</b>                        | Keine   |  |            |           |
| <b>Prüfung</b>                                | Klausur   |  |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 120 Minuten   |  |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Engineering Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Kernqualifikation: Pflicht |  |            |           |

| <b>Lehrveranstaltung L0647: Mathematical Analysis</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 4  |
| <b>LP</b>   | 4  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 64, Präsenzstudium 56   |
| <b>Dozenten</b>                                       | Dr. Julian Großmann  |
| <b>Sprachen</b>                                       | EN   |
| <b>Zeitraum</b>                                       | SoSe   |
| <b>Inhalt</b>   | Convergence, sequences, and series<br>Continuity<br>Elementary functions<br>Differential calculus<br>Integral calculus<br>Sequences of functions |
| <b>Literatur</b>                                      | Königsberger: Analysis<br>Forster: Analysis  |

| <b>Lehrveranstaltung L0648: Mathematical Analysis</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Hörsaalübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                                       | Dr. Julian Großmann                |
| <b>Sprachen</b>                                       | EN                                 |
| <b>Zeitraum</b>                                       | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                      | Siehe korrespondierende Vorlesung  |

| <b>Lehrveranstaltung L0649: Mathematical Analysis</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                                       | Dr. Julian Großmann                |
| <b>Sprachen</b>                                       | EN                                 |
| <b>Zeitraum</b>                                       | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                      | Siehe korrespondierende Vorlesung  |

## Modul M0829: Grundlagen der Betriebswirtschaftslehre

### Lehrveranstaltungen

| Titel   | Typ          | SWS | LP |
|---|--------------|-----|----|
| Betriebswirtschaftliche Übung (L0882)           | Gruppenübung | 2   | 3  |
| Grundlagen der Betriebswirtschaftslehre (L0880) | Vorlesung    | 3   | 3  |

|   |  |
|---|--|
| <b>Modulverantwortlicher</b>                  | Prof. Christoph Ihl  |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |
| <b>Empfohlene Vorkenntnisse</b>               | Schulkenntnisse in Mathematik und Wirtschaft   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                          | <p>Die Studierenden können...</p> <ul style="list-style-type: none"> <li>• grundlegende Begriffe und Kategorien aus dem Bereich Wirtschaft und Management benennen und erklären</li> <li>• grundlegende Aspekte wettbewerblichen Unternehmertums beschreiben (Betrieb und Unternehmung, betrieblicher Zielbildungsprozess)</li> <li>• wesentliche betriebliche Funktionen erläutern, insb. Funktionen der Wertschöpfungskette (z.B. Produktion und Beschaffung, Innovationsmanagement, Absatz und Marketing) sowie Querschnittsfunktionen (z.B. Organisation, Personalmanagement, Supply Chain Management, Informationsmanagement) und die wesentlichen Aspekte von Entrepreneurship-Projekten benennen</li> <li>• Grundlagen der Unternehmensplanung (Entscheidungstheorie, Planung und Kontrolle) wie auch spezielle Planungsaufgaben (z.B. Projektplanung, Investition und Finanzierung) erläutern</li> <li>• Grundlagen des Rechnungswesens erklären (Buchführung, Bilanzierung, Kostenrechnung, Controlling)</li> </ul> |
| <i>Wissen</i>                                 |  |
| <b>Fertigkeiten</b>                           | <p>Die Studierenden können</p> <ul style="list-style-type: none"> <li>• Unternehmensziele definieren und in ein Zielsystem einordnen sowie Zielsysteme strukturieren</li> <li>• Organisations- und Personalstrukturen von Unternehmen analysieren</li> <li>• Methoden für Entscheidungsprobleme unter mehrfacher Zielsetzung, unter Ungewissheit sowie unter Risiko zur Lösung von entsprechenden Problemen anwenden</li> <li>• Produktions- und Beschaffungssysteme sowie betriebliche Informationssysteme analysieren und einordnen</li> <li>• Einfache preispolitische und weitere Instrumente des Marketing analysieren und anwenden</li> <li>• Grundlegende Methoden der Finanzmathematik auf Investitions- und Finanzierungsprobleme anwenden</li> <li>• Die Grundlagen der Buchhaltung, Bilanzierung, Kostenrechnung und des Controlling erläutern und Methoden aus diesen Bereichen auf einfache Problemstellungen anwenden.</li> </ul>  |
| <i>Fertigkeiten</i>                           |  |
| <b>Personale Kompetenzen</b>                  | <p>Die Studierenden sind in der Lage</p> <ul style="list-style-type: none"> <li>• sich im Team zu organisieren und ein Projekt aus dem Bereich Entrepreneurship gemeinsam zu bearbeiten und einen Projektbericht zu erstellen</li> <li>• erfolgreich problemlösungsorientiert zu kommunizieren</li> <li>• respektvoll und erfolgreich zusammenzuarbeiten</li> </ul>  |
| <i>Sozialkompetenz</i>                        |  |
| <b>Selbstständigkeit</b>                      | <p>Die Studierenden sind in der Lage</p> <ul style="list-style-type: none"> <li>• Ein Projekt in einem Team zu bearbeiten und einer Lösung zuzuführen</li> <li>• unter Anleitung einen Projektbericht zu verfassen</li> </ul>  |
| <i>Selbstständigkeit</i>                      |  |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70  |

|   |  |
|---|--|
| <b>Leistungspunkte</b>                  | 6  |
| <b>Studienleistung</b>                  | Keine  |
| <b>Prüfung</b>                          | Fachtheoretisch-fachpraktische Arbeit  |
| <b>Prüfungsdauer und -umfang</b>        | mehrere schriftliche Leistungen über das Semester verteilt   |
| <b>Zuordnung zu folgenden Curricula</b> | <p>Allgemeine Ingenieurwissenschaften (7 Semester): Kernqualifikation: Pflicht<br/>                 Bau- und Umweltingenieurwesen: Kernqualifikation: Pflicht<br/>                 Bau- und Umweltingenieurwesen: Vertiefung Bauingenieurwesen: Wahlpflicht<br/>                 Bau- und Umweltingenieurwesen: Vertiefung Wasser und Umwelt: Wahlpflicht<br/>                 Bau- und Umweltingenieurwesen: Vertiefung Verkehr und Mobilität: Wahlpflicht<br/>                 Bioverfahrenstechnik: Kernqualifikation: Pflicht<br/>                 Computer Science: Kernqualifikation: Pflicht<br/>                 Data Science: Kernqualifikation: Pflicht<br/>                 Elektrotechnik: Kernqualifikation: Pflicht<br/>                 Energie- und Umwelttechnik: Kernqualifikation: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Bauingenieurwesen: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Energie- und Umwelttechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Schiffbau: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Verfahrenstechnik: Pflicht<br/>                 General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br/>                 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br/>                 Logistik und Mobilität: Kernqualifikation: Pflicht<br/>                 Maschinenbau: Kernqualifikation: Pflicht<br/>                 Mechatronik: Kernqualifikation: Pflicht<br/>                 Orientierungsstudium: Kernqualifikation: Wahlpflicht<br/>                 Schiffbau: Kernqualifikation: Pflicht<br/>                 Technomathematik: Kernqualifikation: Pflicht<br/>                 Verfahrenstechnik: Kernqualifikation: Pflicht</p> |

| <b>Lehrveranstaltung L0882: Betriebswirtschaftliche Übung</b> |   |
|---|---|
| <b>Typ</b>  | Gruppenübung  |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>                              | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>   | Prof. Christoph Ihl, Katharina Roedelius, Tobias Vlcek  |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | WiSe/SoSe   |
| <b>Inhalt</b>   | <p>In der betriebswirtschaftlichen Horsaalübung werden die Inhalte der Vorlesung durch praktische Beispiele und die Anwendung der diskutierten Werkzeuge vertieft.</p> <p>Bei angemessener Nachfrage wird parallel auch eine Problemorientierte Lehrveranstaltung angeboten, die Studierende alternativ wählen können. Hier bearbeiten die Studierenden in Gruppen ein selbstgewähltes Projekt, das sich thematisch mit der Ausarbeitung einer innovativen Geschäftsidee aus Sicht eines etablierten Unternehmens oder Startups befasst. Auch hier sollen die betriebswirtschaftlichen Grundkenntnisse aus der Vorlesung zum praktischen Einsatz kommen. Die Gruppenarbeit erfolgt unter Anleitung eines Mentors.</p> |
| <b>Literatur</b>  | Relevante Literatur aus der korrespondierenden Vorlesung.   |

| <b>Lehrveranstaltung L0880: Grundlagen der Betriebswirtschaftslehre</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 3  |
| <b>LP</b>   | 3  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 48, Präsenzstudium 42   |
| <b>Dozenten</b>   | Prof. Christoph Ihl, Prof. Thorsten Blecker, Prof. Christian Lüthje, Prof. Christian Ringle, Prof. Kathrin Fischer, Prof. Cornelius Herstatt, Prof. Wolfgang Kersten, Prof. Matthias Meyer, Prof. Thomas Wrona   |
| <b>Sprachen</b>   | DE   |
| <b>Zeitraum</b>   | WiSe/SoSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Die Abgrenzung der BWL von der VWL und die Gliederungsmöglichkeiten der BWL</li> <li>• Wichtige Definitionen aus dem Bereich Management und Wirtschaft</li> <li>• Die wichtigsten Unternehmensziele und ihre Einordnung sowie (Kern-) Funktionen der Unternehmung</li> <li>• Die Bereiche Produktion und Beschaffungsmanagement, der Begriff des Supply Chain Management und die Bestandteile einer Supply Chain</li> <li>• Die Definition des Begriffs Information, die Organisation des Informations- und Kommunikations (IuK)-Systems und Aspekte der Datensicherheit; Unternehmensstrategie und strategische Informationssysteme</li> <li>• Der Begriff und die Bedeutung von Innovationen, insbesondere Innovationschancen, -risiken und prozesse</li> <li>• Die Bedeutung des Marketing, seine Aufgaben, die Abgrenzung von B2B- und B2C-Marketing</li> <li>• Aspekte der Marketingforschung (Marktportfolio, Szenario-Technik) sowie Aspekte der strategischen und der operativen Planung und Aspekte der Preispolitik</li> <li>• Die grundlegenden Organisationsstrukturen in Unternehmen und einige Organisationsformen</li> <li>• Grundzüge des Personalmanagements</li> <li>• Die Bedeutung der Planung in Unternehmen und die wesentlichen Schritte eines Planungsprozesses</li> <li>• Die wesentlichen Bestandteile einer Entscheidungssituation sowie Methoden für Entscheidungsprobleme unter mehrfacher Zielsetzung, unter Ungewissheit sowie unter Risiko</li> <li>• Grundlegende Methoden der Finanzmathematik</li> <li>• Die Grundlagen der Buchhaltung, der Bilanzierung und der Kostenrechnung</li> <li>• Die Bedeutung des Controlling im Unternehmen und ausgewählte Methoden des Controlling</li> <li>• Die wesentlichen Aspekte von Entrepreneurship-Projekten</li> </ul> <p>Neben der Vorlesung, die die Fachinhalte vermittelt, erarbeiten die Studierenden selbstständig in Gruppen einen Business-Plan für ein Gründungsprojekt. Dafür wird auch das wissenschaftliche Arbeiten und Schreiben gezielt unterstützt.</p> |
| <b>Literatur</b>  | <p>Bamberg, G., Coenenberg, A.: Betriebswirtschaftliche Entscheidungslehre, 14. Aufl., München 2008</p> <p>Eisenführ, F., Weber, M.: Rationales Entscheiden, 4. Aufl., Berlin et al. 2003</p> <p>Heinhold, M.: Buchführung in Fallbeispielen, 10. Aufl., Stuttgart 2006.</p> <p>Kruschwitz, L.: Finanzmathematik. 3. Auflage, München 2001.</p> <p>Pellens, B., Fülber, R. U., Gassen, J., Sellhorn, T.: Internationale Rechnungslegung, 7. Aufl., Stuttgart 2008.</p> <p>Schweitzer, M.: Planung und Steuerung, in: Bea/Friedl/Schweitzer: Allgemeine Betriebswirtschaftslehre, Bd. 2: Führung, 9. Aufl., Stuttgart 2005.</p> <p>Weber, J., Schäffer, U. : Einführung in das Controlling, 12. Auflage, Stuttgart 2008.</p> <p>Weber, J./Weißberger, B.: Einführung in das Rechnungswesen, 7. Auflage, Stuttgart 2006.</p>   |

| Modul M1432: Programmierparadigmen            |  |                |            |           |
|---|--|----------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |                |            |           |
| <b>Titel</b>                                  |  | <b>Typ</b>     | <b>SWS</b> | <b>LP</b> |
| Programmierparadigmen (L2169)                 |  | Vorlesung      | 2          | 2         |
| Programmierparadigmen (L2170)                 |  | Hörsaalübung   | 1          | 1         |
| Programmierparadigmen (L2171)                 |  | Laborpraktikum | 2          | 3         |
| <b>Modulverantwortlicher</b>                  | NN   |                |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |                |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | Veranstaltung Prozedurale Programmierung oder gleichwertige Programmierkenntnisse in imperativer Programmierung  |                |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |                |            |           |
| <b>Fachkompetenz</b>                          | Die Studenten haben ein grundlegendes Verständnis über die objektorientierte und die generische Programmierung erworben und können diese in eigenen Programmierprojekten umsetzen. Sie können eigene Klassenhierarchien erstellen und verschiedene Formen der Vererbung unterscheiden. Sie haben ein grundlegendes Verständnis des Polymorphismus und können zwischen Laufzeit- und Compilierzeit-Polymorphismus unterscheiden. Die Studenten sind mit dem Konzept der Datenkapselung vertraut und können Schnittstellen in private und öffentliche Methoden unterteilen. Sie können mit Exceptions umgehen und nutzen generische Programmierung um Datenstrukturen zu verallgemeinern. Die Studenten können die Vor- und Nachteile der beiden Programmierparadigmen |                |            |           |
| <i>Wissen</i>                                 |  |                |            |           |
| <i>Fertigkeiten</i>                           | Die Studenten können eine mittelgroße Problemstellung in Teilprobleme zerlegen und darauf aufbauend eigene Klassen in einer objektorientierten Programmiersprache erstellen. Sie können dabei ein öffentliche und private Schnittstellen entwerfen und die Implementierung durch Abstraktion generisch und erweiterbar umsetzen. Sie können verschiedene Sprachkonstrukte einer modernen Programmiersprache unterscheiden und diese geeignet in der Implementierung nutzen. Sie können Unit Tests entwerfen und implementieren.  |                |            |           |
| <b>Personale Kompetenzen</b>                  |  |                |            |           |
| <i>Sozialkompetenz</i>                        | Studierende können in Teams arbeiten und in Foren kommunizieren.   |                |            |           |
| <i>Selbstständigkeit</i>                      | In Programmierpraktikum lernen die Studenten unter Aufsicht die objektorientierte Programmierung. In Übungen entwickeln sie individuell und unabhängig Lösungen und erhalten Feedback.   |                |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70  |                |            |           |
| <b>Leistungspunkte</b>                        | 6  |                |            |           |
| <b>Studienleistung</b>                        | Keine  |                |            |           |
| <b>Prüfung</b>                                | Klausur  |                |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |                |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht  |                |            |           |

| <b>Lehrveranstaltung L2169: Programmierparadigmen</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 2   |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 32, Präsenzstudium 28  |
| <b>Dozenten</b>                                       | Dozenten des SD E   |
| <b>Sprachen</b>                                       | DE/EN   |
| <b>Zeitraum</b>                                       | SoSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Grundlegende Idee der Objektorientierten Programmierung</li> <li>• Klassen und Objekte</li> <li>• Vererbung (einfach, mehrfach)</li> <li>• Schnittstellen (Interfaces)</li> <li>• Datenkapselung (private / public usw.)</li> <li>• Ausnahmebehandlung (Exceptions)</li> <li>• Generische Programmierung und deren Umsetzung im Compiler</li> <li>• Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen</li> </ul> |
| <b>Literatur</b>                                      | Skript  |

| <b>Lehrveranstaltung L2170: Programmierparadigmen</b> |   |
|---|---|
| <b>Typ</b>  | Hörsaalübung  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 16, Präsenzstudium 14  |
| <b>Dozenten</b>                                       | Dozenten des SD E   |
| <b>Sprachen</b>                                       | DE/EN   |
| <b>Zeitraum</b>                                       | SoSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Grundlegende Idee der Objektorientierten Programmierung</li> <li>• Klassen und Objekte</li> <li>• Vererbung (einfach, mehrfach)</li> <li>• Schnittstellen (Interfaces)</li> <li>• Datenkapselung (private / public usw.)</li> <li>• Ausnahmebehandlung (Exceptions)</li> <li>• Generische Programmierung und deren Umsetzung im Compiler</li> <li>• Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen</li> </ul> |
| <b>Literatur</b>                                      | Skript  |

| <b>Lehrveranstaltung L2171: Programmierparadigmen</b> |   |
|---|---|
| <b>Typ</b>  | Laborpraktikum  |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>                                       | Dozenten des SD E   |
| <b>Sprachen</b>                                       | DE/EN   |
| <b>Zeitraum</b>                                       | SoSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Grundlegende Idee der Objektorientierten Programmierung</li> <li>• Klassen und Objekte</li> <li>• Vererbung (einfach, mehrfach)</li> <li>• Schnittstellen (Interfaces)</li> <li>• Datenkapselung (private / public usw.)</li> <li>• Ausnahmebehandlung (Exceptions)</li> <li>• Generische Programmierung und deren Umsetzung im Compiler</li> <li>• Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen</li> </ul> |
| <b>Literatur</b>                                      | Skript  |

| Modul M0730: Technische Informatik            |  |               |  |
|---|--|---------------|--|
| <b>Lehrveranstaltungen</b>                    |  |               |  |
| <b>Titel</b>                                  |  | <b>Typ</b>    | <b>SWS</b> <b>LP</b>                               |
| Technische Informatik (L0321)                 |  | Vorlesung     | 3            4                                     |
| Technische Informatik (L0324)                 |  | Gruppenübung  | 1            2                                     |
| <b>Modulverantwortlicher</b>                  | Prof. Heiko Falk   |               |  |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |               |  |
| <b>Empfohlene Vorkenntnisse</b>               | Grundkenntnisse der Elektrotechnik   |               |  |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |               |  |
| <b>Fachkompetenz</b>                          | <p>Dieses Modul vermittelt Grundkenntnisse der Funktionsweise von Rechensystemen. Abgedeckt werden die Ebenen von der Assemblerprogrammierung bis zur Gatterebene. Das Modul behandelt folgende Inhalte:</p> <ul style="list-style-type: none"> <li>• Einführung</li> <li>• Kombinatorische Logik: Gatter, Boolesche Algebra, Schaltfunktionen, Synthese von Schaltungen, Schaltnetze</li> <li>• Sequentielle Logik: Flip-Flops, Schaltwerke, systematischer Schaltwerkdentwurf</li> <li>• Technologische Grundlagen</li> <li>• Rechnerarithmetik: Ganzzahlige Addition, Subtraktion, Multiplikation und Division</li> <li>• Grundlagen der Rechnerarchitektur: Programmiermodelle, MIPS-Einzelzyklusmaschine, Pipelining</li> <li>• Speicher-Hardware: Speicherhierarchien, SRAM, DRAM, Caches</li> <li>• Ein-/Ausgabe: I/O aus Sicht der CPU, Prinzipien der Datenübergabe, Point-to-Point Verbindungen, Busse</li> </ul>  |               |  |
| <i>Wissen</i>                                 |  |               |  |
| <i>Fertigkeiten</i>                           | <p>Die Studierenden fassen ein Rechensystem aus der Perspektive des Architekten auf, d.h. sie erkennen die interne Struktur und den physischen Aufbau von Rechensystemen. Die Studierenden können analysieren, wie hochspezifische und individuelle Rechner aus einer Sammlung gängiger Einzelkomponenten zusammengesetzt werden. Sie sind in der Lage, die unterschiedlichen Abstraktionsebenen heutiger Rechensysteme - von Gattern und Schaltungen bis hin zu Prozessoren - zu unterscheiden und zu erklären.</p> <p>Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage, die Wechselwirkungen zwischen einem physischen Rechensystem und der darauf ausgeführten Software beurteilen zu können. Insbesondere sollen sie die Konsequenzen der Ausführung von Software in den hardwarenahen Schichten von der Assemblersprache bis zu Gattern erkennen können. Sie sollen so in die Lage versetzt werden, Auswirkungen unterer Schichten auf die Leistung des Gesamtsystems abzuschätzen und geeignete Optionen vorzuschlagen.</p> |               |  |
| <b>Personale Kompetenzen</b>                  |  |               |  |
| <i>Sozialkompetenz</i>                        | Die Studierenden sind nach Abschluss des Moduls in der Lage, ähnliche Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.  |               |  |
| <i>Selbstständigkeit</i>                      | Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.  |               |  |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |               |  |
| <b>Leistungspunkte</b>                        | 6  |               |  |
| <b>Studienleistung</b>                        | <b>Verpflichtend</b>   | <b>Anteil</b> | <b>Art der Studienleistung</b> <b>Beschreibung</b> |
|   | Ja   | 10 %          | Übungsaufgaben                                     |
| <b>Prüfung</b>                                | Klausur  |               |  |
| <b>Prüfungsdauer und -umfang</b>              | 90 Minuten, Inhalte der Vorlesung und Übungen  |               |  |
|   | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht   |               |  |

|  |   |
|--|---|
| <b>Zuordnung zu folgenden Curricula</b>                  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Schiffbau: Pflicht  |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Elektrotechnik: Pflicht   |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht  |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Energie- und Umwelttechnik: Pflicht   |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Verfahrenstechnik: Pflicht  |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht                                |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht                                |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht                     |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht                 |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht          |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht                             |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht                             |
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Bauingenieurwesen: Pflicht  |
|  | Computer Science: Kernqualifikation: Pflicht  |
|  | Data Science: Kernqualifikation: Wahlpflicht  |
|  | Elektrotechnik: Kernqualifikation: Pflicht  |
|  | General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht  |
|  | General Engineering Science (7 Semester): Vertiefung Bauingenieurwesen: Pflicht   |
|  | General Engineering Science (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht  |
|  | General Engineering Science (7 Semester): Vertiefung Energie- und Umwelttechnik: Pflicht  |
|  | General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht  |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht                                       |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht                                    |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht                            |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht        |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht                                       |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht                 |
|  | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht                        |
|  | General Engineering Science (7 Semester): Vertiefung Schiffbau: Pflicht   |
|  | General Engineering Science (7 Semester): Vertiefung Verfahrenstechnik: Pflicht   |
|  | General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht   |
|  | Informatik-Ingenieurwesen: Kernqualifikation: Pflicht   |
| Mechatronik: Kernqualifikation: Pflicht                  |   |
| Technomathematik: Vertiefung II. Informatik: Wahlpflicht |   |

| <b>Lehrveranstaltung L0321: Technische Informatik</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 3   |
| <b>LP</b>   | 4   |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 78, Präsenzstudium 42  |
| <b>Dozenten</b>                                       | Prof. Heiko Falk  |
| <b>Sprachen</b>                                       | DE/EN   |
| <b>Zeitraum</b>                                       | WiSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Einführung</li> <li>• Kombinatorische Logik</li> <li>• Sequentielle Logik</li> <li>• Technologische Grundlagen</li> <li>• Zahlendarstellungen und Rechnerarithmetik</li> <li>• Grundlagen der Rechnerarchitektur</li> <li>• Speicher-Hardware</li> <li>• Ein-/Ausgabe</li> </ul> |
| <b>Literatur</b>                                      | <ul style="list-style-type: none"> <li>• A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> </ul>         |

| <b>Lehrveranstaltung L0324: Technische Informatik</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 1                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 46, Präsenzstudium 14 |
| <b>Dozenten</b>                                       | Prof. Heiko Falk                   |
| <b>Sprachen</b>                                       | DE/EN                              |
| <b>Zeitraum</b>                                       | WiSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                      | Siehe korrespondierende Vorlesung  |

| Modul M0834: Computernetworks and Internet Security |   |              |                      |
|---|---|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                          |   |              |                      |
| <b>Titel</b>  |   | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Rechnernetze und Internet-Sicherheit (L1098)        |   | Vorlesung    | 3            5       |
| Rechnernetze und Internet-Sicherheit (L1099)        |   | Gruppenübung | 1            1       |
| <b>Modulverantwortlicher</b>                        | Prof. Andreas Timm-Giel   |              |                      |
| <b>Zulassungsvoraussetzungen</b>                    | None  |              |                      |
| <b>Empfohlene Vorkenntnisse</b>                     | Basics of Computer Science  |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b>       | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |              |                      |
| <b>Fachkompetenz</b>                                | <p><i>Wissen</i><br/>Students are able to explain important and common Internet protocols in detail and classify them, in order to be able to analyse and develop networked systems in further studies and job.</p> <p><i>Fertigkeiten</i><br/>Students are able to analyse common Internet protocols and evaluate the use of them in different domains.</p> <p><b>Personale Kompetenzen</b></p> <p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i><br/>Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand it.</p> |              |                      |
|   |   |              |                      |
|   |   |              |                      |
|   |   |              |                      |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 124, Präsenzstudium 56   |              |                      |
| <b>Leistungspunkte</b>                              | 6   |              |                      |
| <b>Studienleistung</b>                              | Keine   |              |                      |
| <b>Prüfung</b>                                      | Klausur   |              |                      |
| <b>Prüfungsdauer und -umfang</b>                    | 120 min   |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>             | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Wahlpflicht<br>Elektrotechnik: Kernqualifikation: Wahlpflicht<br>Engineering Science: Vertiefung Mechatronics: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht         |              |                      |

| Lehrveranstaltung L1098: Computer Networks and Internet Security |   |
|--|---|
| <b>Typ</b>   | Vorlesung   |
| <b>SWS</b>   | 3   |
| <b>LP</b>  | 5   |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 108, Präsenzstudium 42   |
| <b>Dozenten</b>  | Prof. Andreas Timm-Giel, Prof. Dieter Gollmann  |
| <b>Sprachen</b>  | EN  |
| <b>Zeitraum</b>  | WiSe  |
| <b>Inhalt</b>  | <p>In this class an introduction to computer networks with focus on the Internet and its security is given. Basic functionality of complex protocols are introduced. Students learn to understand these and identify common principles. In the exercises these basic principles and an introduction to performance modelling are addressed using computing tasks and (virtual) labs.</p> <p>In the second part of the lecture an introduction to Internet security is given.</p> <p>This class comprises:</p> <ul style="list-style-type: none"> <li>• Application layer protocols (HTTP, FTP, DNS)</li> <li>• Transport layer protocols (TCP, UDP)</li> <li>• Network Layer (Internet Protocol, routing in the Internet)</li> <li>• Data link layer with media access at the example of Ethernet</li> <li>• Multimedia applications in the Internet</li> <li>• Network management</li> <li>• Internet security: IPSec</li> <li>• Internet security: Firewalls</li> </ul> |
| <b>Literatur</b>   | <ul style="list-style-type: none"> <li>• Kurose, Ross, Computer Networking - A Top-Down Approach, 6th Edition, Addison-Wesley</li> <li>• Kurose, Ross, Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium; Auflage: 6. Auflage</li> <li>• W. Stallings: Cryptography and Network Security: Principles and Practice, 6th edition</li> </ul> <p>Further literature is announced at the beginning of the lecture.</p>  |

| Lehrveranstaltung L1099: Computer Networks and Internet Security |  |
|--|--|
| <b>Typ</b>   | Gruppenübung                                   |
| <b>SWS</b>   | 1  |
| <b>LP</b>  | 1  |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 16, Präsenzstudium 14             |
| <b>Dozenten</b>  | Prof. Andreas Timm-Giel, Prof. Dieter Gollmann |
| <b>Sprachen</b>  | EN   |
| <b>Zeitraum</b>  | WiSe   |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung              |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung              |

| Modul M1423: Algorithmen und Datenstrukturen  |   |              |                      |
|---|---|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                    |   |              |                      |
| <b>Titel</b>                                  |   | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Algorithmen und Datenstrukturen (L2046)       |   | Vorlesung    | 4            4       |
| Algorithmen und Datenstrukturen (L2047)       |   | Gruppenübung | 1            2       |
| <b>Modulverantwortlicher</b>                  | Prof. Matthias Mnich  |              |                      |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |              |                      |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Diskrete Algebraische Strukturen</li> <li>• Mathematik I</li> <li>• Mathematik II</li> <li>• Prozedurale Programming</li> <li>• Objectorientierte Programmierung</li> </ul>  |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |              |                      |
| <b>Fachkompetenz</b>                          |   |              |                      |
| <i>Wissen</i>                                 | <ul style="list-style-type: none"> <li>• Studierende können die grundlegenden Begriffe des Algorithmenentwurfs, der Algorithmenanalyse und Problemreduktionen benennen und anhand von Beispielen erklären.</li> <li>• Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</li> <li>• Sie kennen Beweisstrategien und können diese wiedergeben.</li> </ul>   |              |                      |
| <i>Fertigkeiten</i>                           | <ul style="list-style-type: none"> <li>• Studierende können diskrete Entscheidungsprobleme, Such- und Optimierungsprobleme mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.</li> <li>• Studierende sind in der Lage, sich weitere einfache logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.</li> <li>• Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.</li> </ul> |              |                      |
| <b>Personale Kompetenzen</b>                  |   |              |                      |
| <i>Sozialkompetenz</i>                        | <ul style="list-style-type: none"> <li>• Studierende sind in der Lage, in Teams zusammenzuarbeiten und die Mathematik als gemeinsame Sprache zu entdecken und beherrschen.</li> <li>• Sie können sich dabei insbesondere gegenseitig neue Konzepte erklären und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen.</li> </ul>  |              |                      |
| <i>Selbstständigkeit</i>                      | <ul style="list-style-type: none"> <li>• Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen.</li> <li>• Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume an schwierigen Problemstellungen zu arbeiten.</li> </ul>   |              |                      |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70   |              |                      |
| <b>Leistungspunkte</b>                        | 6   |              |                      |
| <b>Studienleistung</b>                        | Keine   |              |                      |
| <b>Prüfung</b>                                | Klausur   |              |                      |
| <b>Prüfungsdauer und -umfang</b>              | 60 min  |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht   |              |                      |

| <b>Lehrveranstaltung L2046: Algorithmen und Datenstrukturen</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 4   |
| <b>LP</b>   | 4   |
| <b>Arbeitsaufwand in Stunden</b>                                | Eigenstudium 64, Präsenzstudium 56  |
| <b>Dozenten</b>   | Prof. Matthias Mnich  |
| <b>Sprachen</b>   | DE/EN   |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Sortieren durch Einfügen</li> <li>• Registermaschinen</li> <li>• Asymptotische Analyse, Landau Notation</li> <li>• Polynomialzeit Algorithmen and NP-Vollständigkeit</li> <li>• Divide-and-conquer, Merge sort</li> <li>• Strassens Algorithmus</li> <li>• Greedy Algorithmen</li> <li>• Dynamische Programmierung</li> <li>• Quicksort</li> <li>• AVL-trees, B-trees</li> <li>• Hashing</li> <li>• Tiefensuche und Breitensuche</li> <li>• Kürzeste Wege</li> <li>• Fluss Probleme, Ford-Fulkerson Algorithmus</li> </ul> |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013</li> <li>• S. Skiena: The Algorithm Design Manual. Springer, 2008</li> <li>• J. M. Kleinberg and É. Tardos. Algorithm Design. Addison-Wesley, 2005.</li> </ul>  |

| <b>Lehrveranstaltung L2047: Algorithmen und Datenstrukturen</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 1                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                                | Eigenstudium 46, Präsenzstudium 14 |
| <b>Dozenten</b>   | Prof. Matthias Mnich               |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | WiSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

## Modul M0853: Mathematik III

### Lehrveranstaltungen

| Titel   | Typ          | SWS | LP |
|---|--------------|-----|----|
| Analysis III (L1028)  | Vorlesung    | 2   | 2  |
| Analysis III (L1029)  | Gruppenübung | 1   | 1  |
| Analysis III (L1030)  | Hörsaalübung | 1   | 1  |
| Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (L1031) | Vorlesung    | 2   | 2  |
| Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (L1032) | Gruppenübung | 1   | 1  |
| Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (L1033) | Hörsaalübung | 1   | 1  |

|                              |                    |
|------------------------------|--------------------|
| <b>Modulverantwortlicher</b> | Prof. Anusch Taraz |
|------------------------------|--------------------|

|                                  |       |
|----------------------------------|-------|
| <b>Zulassungsvoraussetzungen</b> | Keine |
|----------------------------------|-------|

|                                 |                   |
|---------------------------------|-------------------|
| <b>Empfohlene Vorkenntnisse</b> | Mathematik I + II |
|---------------------------------|-------------------|

|   |   |
|---|---|
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht |
|---|---|

| Fachkompetenz            |  |
|--------------------------|--|
| <i>Wissen</i>            | <ul style="list-style-type: none"> <li>Studierende können die grundlegenden Begriffe aus dem Gebiet der Analysis und Differentialgleichungen benennen und anhand von Beispielen erklären.</li> <li>Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</li> <li>Sie kennen Beweisstrategien und können diese wiedergeben.</li> </ul>   |
| <i>Fertigkeiten</i>      | <ul style="list-style-type: none"> <li>Studierende können Aufgabenstellungen aus dem Gebiet der Analysis und Differentialgleichungen mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.</li> <li>Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbständig zu erschließen und können diese verifizieren.</li> <li>Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.</li> </ul> |
| Personale Kompetenzen    |  |
| <i>Sozialkompetenz</i>   | <ul style="list-style-type: none"> <li>Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache.</li> <li>Sie können dabei insbesondere neue Konzepte adressatengerecht kommunizieren und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen.</li> </ul>  |
| <i>Selbstständigkeit</i> | <ul style="list-style-type: none"> <li>Studierende können eigenständig ihr Verständnis komplexer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen.</li> <li>Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume zielgerichtet an schwierigen Problemstellungen zu arbeiten.</li> </ul>   |

|                                  |                                      |
|----------------------------------|--------------------------------------|
| <b>Arbeitsaufwand in Stunden</b> | Eigenstudium 128, Präsenzstudium 112 |
|----------------------------------|--------------------------------------|

|                        |   |
|------------------------|---|
| <b>Leistungspunkte</b> | 8 |
|------------------------|---|

|                        |       |
|------------------------|-------|
| <b>Studienleistung</b> | Keine |
|------------------------|-------|

|                |         |
|----------------|---------|
| <b>Prüfung</b> | Klausur |
|----------------|---------|

|                                  |  |
|----------------------------------|--|
| <b>Prüfungsdauer und -umfang</b> | 60 min (Analysis III) + 60 min (Differentialgleichungen 1) |
|----------------------------------|--|

|  |  |
|--|--|
|  | Allgemeine Ingenieurwissenschaften (7 Semester): Kernqualifikation: Pflicht<br>Bau- und Umweltingenieurwesen: Kernqualifikation: Pflicht<br>Bioverfahrenstechnik: Kernqualifikation: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Digitaler Maschinenbau: Kernqualifikation: Pflicht<br>Elektrotechnik: Kernqualifikation: Pflicht |
|--|--|

|   |   |
|---|---|
| <b>Zuordnung zu folgenden Curricula</b> | Energie- und Umwelttechnik: Kernqualifikation: Pflicht<br>Engineering Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Kernqualifikation: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Maschinenbau: Kernqualifikation: Pflicht<br>Mechatronik: Kernqualifikation: Pflicht<br>Schiffbau: Kernqualifikation: Pflicht<br>Verfahrenstechnik: Kernqualifikation: Pflicht |
|---|---|

| Lehrveranstaltung L1028: Analysis III |  |
|---------------------------------------|--|
| <b>Typ</b>                            | Vorlesung  |
| <b>SWS</b>                            | 2  |
| <b>LP</b>                             | 2  |
| <b>Arbeitsaufwand in Stunden</b>      | Eigenstudium 32, Präsenzstudium 28   |
| <b>Dozenten</b>                       | Dozenten des Fachbereiches Mathematik der UHH  |
| <b>Sprachen</b>                       | DE   |
| <b>Zeitraum</b>                       | WiSe   |
| <b>Inhalt</b>                         | Grundzüge der Differential- und Integralrechnung mehrerer Variablen: <ul style="list-style-type: none"> <li>• Differentialrechnung mehrerer Veränderlichen</li> <li>• Mittelwertsätze und Taylorscher Satz</li> <li>• Extremwertbestimmung</li> <li>• Implizit definierte Funktionen</li> <li>• Extremwertbestimmung bei Gleichungsnebenbedingungen</li> <li>• Newton-Verfahren für mehrere Variablen</li> <li>• Bereichsintegrale</li> <li>• Kurven- und Flächenintegrale</li> <li>• Integralsätze von Gauß und Stokes</li> </ul> |
| <b>Literatur</b>                      | <ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>  |

| Lehrveranstaltung L1029: Analysis III |   |
|---------------------------------------|---|
| <b>Typ</b>                            | Gruppenübung                                  |
| <b>SWS</b>                            | 1   |
| <b>LP</b>                             | 1   |
| <b>Arbeitsaufwand in Stunden</b>      | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>                       | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>                       | DE  |
| <b>Zeitraum</b>                       | WiSe  |
| <b>Inhalt</b>                         | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>                      | Siehe korrespondierende Vorlesung             |

| Lehrveranstaltung L1030: Analysis III |   |
|---------------------------------------|---|
| <b>Typ</b>                            | Hörsaalübung                                  |
| <b>SWS</b>                            | 1   |
| <b>LP</b>                             | 1   |
| <b>Arbeitsaufwand in Stunden</b>      | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>                       | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>                       | DE  |
| <b>Zeitraum</b>                       | WiSe  |
| <b>Inhalt</b>                         | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>                      | Siehe korrespondierende Vorlesung             |

| <b>Lehrveranstaltung L1031: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 2  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 32, Präsenzstudium 28   |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH  |
| <b>Sprachen</b>   | DE   |
| <b>Zeitraum</b>   | WiSe   |
| <b>Inhalt</b>   | <p>Grundzüge der Theorie und Numerik gewöhnlicher Differentialgleichungen</p> <ul style="list-style-type: none"> <li>• Einführung und elementare Methoden</li> <li>• Existenz und Eindeutigkeit bei Anfangswertaufgaben</li> <li>• Lineare Differentialgleichungen</li> <li>• Stabilität und qualitatives Lösungsverhalten</li> <li>• Randwertaufgaben und Grundbegriffe der Variationsrechnung</li> <li>• Eigenwertaufgaben</li> <li>• Numerische Verfahren zur Integration von Anfangs- und Randwertaufgaben</li> <li>• Grundtypen bei partiellen Differentialgleichungen</li> </ul> |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>  |

| <b>Lehrveranstaltung L1032: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)</b> |   |
|---|---|
| <b>Typ</b>  | Gruppenübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung             |

| <b>Lehrveranstaltung L1033: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)</b> |   |
|---|---|
| <b>Typ</b>  | Hörsaalübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung             |

| Modul M0562: Berechenbarkeit und Komplexität  |  |              |            |           |
|---|--|--------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |              |            |           |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Berechenbarkeit und Komplexität (L0166)       |  | Vorlesung    | 2          | 3         |
| Berechenbarkeit und Komplexität (L0167)       |  | Gruppenübung | 2          | 3         |
| <b>Modulverantwortlicher</b>                  | Prof. Karl-Heinz Zimmermann  |              |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |              |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | Diskrete Algebraische Strukturen sowie Automatentheorie, Logik und Formale Sprachen.   |              |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |            |           |
| <b>Fachkompetenz</b>                          | <p>Wissen: Die Studierenden kennen</p> <ul style="list-style-type: none"> <li>• maschinennahe Modelle der Berechenbarkeit;</li> <li>• abstrakte funktionale Modelle der Berechenbarkeit;</li> <li>• das Konzept der universellen Berechenbarkeit und seine Beschreibung durch partiell-rekursive Funktionen;</li> <li>• das Konzept der Gödelisierung von Berechnungen sowie die Sätze von Kleene, Rice und Rice-Shapiro;</li> <li>• die Konzepte der entscheidbaren und semientscheidbaren Probleme;</li> <li>• die Wortprobleme in Semi-Thue-Systemen, Thue-Systemen, Halbgruppen und Post-Korrespondenz-Systemen;</li> <li>• Hilberts zehntes Problem;</li> <li>• die Komplexitätsklassen P und NP und deren Unterscheidung;</li> <li>• das Konzept der NP-Vollständigkeit sowie den Satz von Cook.</li> </ul> <p>Fertigkeiten: Die Studierenden können</p> <ul style="list-style-type: none"> <li>• maschinennahe und abstrakte Modelle der Berechenbarkeit beschreiben;</li> <li>• Beziehungen zwischen den einzelnen Berechenbarkeitsbegriffen herstellen;</li> <li>• die grundlegenden Sätze von Kleene und Rice rekapitulieren und beweisen;</li> <li>• das Konzept der universellen Berechenbarkeit darlegen;</li> <li>• entscheidbare und semientscheidbare Probleme identifizieren und deren Bezug zu ähnlichen Problemen durch Reduktion herstellen;</li> <li>• die Komplexitätsklassen P und NP beschreiben;</li> <li>• NP-vollständige Probleme lokalisieren.</li> </ul> |              |            |           |
| <i>Wissen</i>                                 |  |              |            |           |
| <i>Fertigkeiten</i>                           |  |              |            |           |
| <b>Personale Kompetenzen</b>                  | <p><i>Sozialkompetenz</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.</p> <p><i>Selbstständigkeit</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachbüchern und anderweitiger Literatur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.</p>   |              |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |              |            |           |
| <b>Leistungspunkte</b>                        | 6  |              |            |           |
| <b>Studienleistung</b>                        | Keine  |              |            |           |
| <b>Prüfung</b>                                | Klausur  |              |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 60 min   |              |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht  |              |            |           |

| <b>Lehrveranstaltung L0166: Berechenbarkeit und Komplexität</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Vorlesung                          |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                                | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | SoSe                               |
| <b>Inhalt</b>   |                                    |
| <b>Literatur</b>  |                                    |

| <b>Lehrveranstaltung L0167: Berechenbarkeit und Komplexität</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                                | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | SoSe                               |
| <b>Inhalt</b>   |                                    |
| <b>Literatur</b>  |                                    |

| Modul M0727: Stochastik                       |  |              |            |           |
|---|--|--------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |              |            |           |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Stochastik (L0777)                            |  | Vorlesung    | 2          | 4         |
| Stochastik (L0778)                            |  | Gruppenübung | 2          | 2         |
| <b>Modulverantwortlicher</b>                  | Prof. Marko Lindner  |              |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |              |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Analysis</li> <li>• Aussagenlogik</li> <li>• Diskrete Algebraische Strukturen (Kombinatorik)</li> </ul>   |              |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |            |           |
| <b>Fachkompetenz</b>                          | Students can explain the main definitions of probability, and they can give basic definitions of modeling elements (random variables, events, dependence, independence assumptions) used in discrete and continuous settings (joint and marginal distributions, density functions). Students can describe characteristic notions such as expected values, variance, standard deviation, and moments.   |              |            |           |
| <i>Wissen</i>                                 | Students can define decision problems and explain algorithms for solving these problems (based on the chain rule or Bayesian networks). Algorithms, or estimators as they are called, can be analyzed in terms of notions such as bias of an estimator, etc. Student can describe the main ideas of stochastic processes and explain algorithms for solving decision and computation problem for stochastic processes. Students can also explain basic statistical detection and estimation techniques.          |              |            |           |
| <i>Fertigkeiten</i>                           | Students can apply algorithms for solving decision problems, and they can justify whether approximation techniques are good enough in various application contexts, i.e., students can derive estimators and judge whether they are applicable or reliable.  |              |            |           |
| <b>Personale Kompetenzen</b>                  |  |              |            |           |
| <i>Sozialkompetenz</i>                        | - Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) and to present their results appropriately (e.g. during exercise class).   |              |            |           |
| <i>Selbstständigkeit</i>                      | - Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.<br>- Students can put their knowledge in relation to the contents of other lectures.<br>- Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.  |              |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |              |            |           |
| <b>Leistungspunkte</b>                        | 6  |              |            |           |
| <b>Studienleistung</b>                        | Keine  |              |            |           |
| <b>Prüfung</b>                                | Klausur  |              |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 120 min  |              |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht<br>Theoretischer Maschinenbau: Kernqualifikation: Wahlpflicht |              |            |           |

| <b>Lehrveranstaltung L0777: Stochastik</b> |   |
|--|---|
| <b>Typ</b>                                 | Vorlesung   |
| <b>SWS</b>                                 | 2   |
| <b>LP</b>                                  | 4   |
| <b>Arbeitsaufwand in Stunden</b>           | Eigenstudium 92, Präsenzstudium 28  |
| <b>Dozenten</b>                            | Dr. Christian Seifert   |
| <b>Sprachen</b>                            | DE/EN   |
| <b>Zeitraum</b>                            | SoSe  |
| <b>Inhalt</b>                              | <p>Grundbegriffe der Wahrscheinlichkeitstheorie</p> <ul style="list-style-type: none"> <li>• Wahrscheinlichkeitsdefinitionen, bedingte Wahrscheinlichkeiten</li> <li>• Zufallsvariablen, Abhängigkeit und Unabhängigkeit von Wahrscheinlichkeiten</li> <li>• Rand- und Verbundwahrscheinlichkeiten</li> <li>• Verteilungs- und Dichtefunktion</li> <li>• Kenngrößen: Erwartungswert, Varianz, Standardabweichung, Momente</li> </ul> <p>Repräsentationsformen für Verbundwahrscheinlichkeiten</p> <ul style="list-style-type: none"> <li>• Bayessche Netzwerke</li> <li>• Semantik, Entscheidungsprobleme, exakte und approximative Algorithmen</li> </ul> <p>Stochastische Prozesse</p> <ul style="list-style-type: none"> <li>• Stationarität und Ergodizität</li> <li>• Korrelationen</li> <li>• Stochastische Signale</li> <li>• Warteschlangen</li> </ul> <p>Detektion &amp; Estimation</p> <ul style="list-style-type: none"> <li>• Detektoren</li> <li>• Schätzregeln und -verfahren</li> <li>• Hypothesen und Verteilungstests</li> <li>• Stochastische Regression</li> </ul> |
| <b>Literatur</b>                           | <ol style="list-style-type: none"> <li>1. Methoden der statistischen Inferenz, Likelihood und Bayes, Held, L., Spektrum 2008</li> <li>2. Stochastik für Informatiker, Dümbgen, L., Springer 2003</li> <li>3. Statistik: Der Weg zur Datenanalyse, Fahrmeir, L., Künstler R., Pigeot, I, Tutz, G., Springer 2010</li> <li>4. Stochastik, Georgii, H.-O., deGruyter, 2009</li> <li>5. Probability and Random Processes, Grimmett, G., Stirzaker, D., Oxford University Press, 2001</li> <li>6. Programmieren mit R, Ligges, U., Springer 2008</li> </ol>  |

| <b>Lehrveranstaltung L0778: Stochastik</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>                                 | Gruppenübung                       |
| <b>SWS</b>                                 | 2                                  |
| <b>LP</b>                                  | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>           | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                            | Dr. Christian Seifert              |
| <b>Sprachen</b>                            | DE/EN                              |
| <b>Zeitraum</b>                            | SoSe                               |
| <b>Inhalt</b>                              | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                           | Siehe korrespondierende Vorlesung  |

| Modul M0732: Software Engineering             |  |               |                                |                     |
|---|--|---------------|--------------------------------|---------------------|
| <b>Lehrveranstaltungen</b>                    |  |               |                                |                     |
| <b>Titel</b>                                  |  | <b>Typ</b>    | <b>SWS</b>                     | <b>LP</b>           |
| Software-Engineering (L0627)                  |  | Vorlesung     | 2                              | 3                   |
| Software-Engineering (L0628)                  |  | Gruppenübung  | 2                              | 3                   |
| <b>Modulverantwortlicher</b>                  | Prof. Sibylle Schupp   |               |                                |                     |
| <b>Zulassungsvoraussetzungen</b>              | None   |               |                                |                     |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Automata theory and formal languages</li> <li>• Procedural programming or Functional programming</li> <li>• Object-oriented programming, algorithms, and data structures</li> </ul>   |               |                                |                     |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |               |                                |                     |
| <b>Fachkompetenz</b>                          | Students explain the phases of the software life cycle, describe the fundamental terminology and concepts of software engineering, and paraphrase the principles of structured software development. They give examples of software-engineering tasks of existing large-scale systems. They write test cases for different test strategies and devise specifications or models using different notations, and critique both. They explain simple design patterns and the major activities in requirements analysis, maintenance, and project planning. |               |                                |                     |
| <i>Wissen</i>                                 |  |               |                                |                     |
| <b>Personale Kompetenzen</b>                  | For a given task in the software life cycle, students identify the corresponding phase and select an appropriate method. They choose the proper approach for quality assurance. They design tests for realistic systems, assess the quality of the tests, and find errors at different levels. They apply and modify non-executable artifacts. They integrate components based on interface specifications.  |               |                                |                     |
| <i>Fertigkeiten</i>                           |  |               |                                |                     |
| <b>Sozialkompetenz</b>                        | Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.  |               |                                |                     |
| <i>Selbstständigkeit</i>                      | Using on-line quizzes and accompanying material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback.   |               |                                |                     |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |               |                                |                     |
| <b>Leistungspunkte</b>                        | 6  |               |                                |                     |
| <b>Studienleistung</b>                        | <b>Verpflichtend</b>   | <b>Anteil</b> | <b>Art der Studienleistung</b> | <b>Beschreibung</b> |
|   | Ja   | 15 %          | Übungsaufgaben                 |                     |
| <b>Prüfung</b>                                | Klausur  |               |                                |                     |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |               |                                |                     |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht  |               |                                |                     |

| <b>Lehrveranstaltung L0627: Software Engineering</b> |  |
|--|--|
| <b>Typ</b>   | Vorlesung  |
| <b>SWS</b>   | 2  |
| <b>LP</b>  | 3  |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 62, Präsenzstudium 28   |
| <b>Dozenten</b>                                      | Prof. Sibylle Schupp   |
| <b>Sprachen</b>                                      | EN   |
| <b>Zeitraum</b>                                      | SoSe   |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Software Life Cycle Models (Waterfall, V-Model, Evolutionary Models, Incremental Models, Iterative Models, Agile Processes)</li> <li>• Requirements (Elicitation Techniques, UML Use Case Diagrams, Functional and Non-Functional Requirements)</li> <li>• Specification (Finite State Machines, Extended FSMs, Petri Nets, Behavioral UML Diagrams, Data Modeling)</li> <li>• Design (Design Concepts, Modules, (Agile) Design Principles)</li> <li>• Object-Oriented Analysis and Design (Object Identification, UML Interaction Diagrams, UML Class Diagrams, Architectural Patterns)</li> <li>• Testing (Blackbox Testing, Whitebox Testing, Control-Flow Testing, Data-Flow Testing, Testing in the Large)</li> <li>• Maintenance and Evolution (Regression Testing, Reverse Engineering, Reengineering)</li> <li>• Project Management (Blackbox Estimation Techniques, Whitebox Estimation Techniques, Project Plans, Gantt Charts, PERT Charts)</li> </ul> |
| <b>Literatur</b>                                     | Kassem A. Saleh, Software Engineering, J. Ross Publishing 2009.  |

| <b>Lehrveranstaltung L0628: Software Engineering</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>                                      | Prof. Sibylle Schupp               |
| <b>Sprachen</b>                                      | EN                                 |
| <b>Zeitraum</b>                                      | SoSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                     | Siehe korrespondierende Vorlesung  |

| Modul M0852: Graphentheorie und Optimierung   |   |              |                      |
|---|---|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                    |   |              |                      |
| <b>Titel</b>                                  |   | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Graphentheorie und Optimierung (L1046)        |   | Vorlesung    | 2            3       |
| Graphentheorie und Optimierung (L1047)        |   | Gruppenübung | 2            3       |
| <b>Modulverantwortlicher</b>                  | Prof. Anusch Taraz  |              |                      |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |              |                      |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Diskrete Algebraische Strukturen</li> <li>• Mathematik I</li> </ul>  |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |              |                      |
| <b>Fachkompetenz</b>                          | <ul style="list-style-type: none"> <li>• Studierende können die grundlegenden Begriffe der Graphentheorie und Optimierung benennen und anhand von Beispielen erklären.</li> <li>• Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</li> <li>• Sie kennen Beweisstrategien und können diese wiedergeben.</li> </ul>   |              |                      |
| <i>Wissen</i>                                 |   |              |                      |
| <i>Fertigkeiten</i>                           | <ul style="list-style-type: none"> <li>• Studierende können Aufgabenstellungen der Graphentheorie und Optimierung mit Hilfe der kennengelernten Konzepte mathematisch modellieren und mit den erlernten Methoden lösen.</li> <li>• Studierende sind in der Lage, sich weitere einfache logische Zusammenhänge zwischen den kennengelernten Konzepten selbständig zu erschließen und können diese verifizieren.</li> <li>• Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.</li> </ul> |              |                      |
| <b>Personale Kompetenzen</b>                  | <ul style="list-style-type: none"> <li>• Studierende sind in der Lage, in heterogen zusammengestellten Teams (mit unterschiedlichem mathematischen Hintergrundwissen und aus unterschiedlichen Studiengängen) zusammenzuarbeiten und die Mathematik als gemeinsame Sprache zu entdecken und beherrschen.</li> <li>• Sie können sich dabei insbesondere gegenseitig neue Konzepte erklären und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen.</li> </ul>  |              |                      |
| <i>Sozialkompetenz</i>                        |   |              |                      |
| <i>Selbstständigkeit</i>                      | <ul style="list-style-type: none"> <li>• Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen.</li> <li>• Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume an schwierigen Problemstellungen zu arbeiten.</li> </ul>   |              |                      |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56   |              |                      |
| <b>Leistungspunkte</b>                        | 6   |              |                      |
| <b>Studienleistung</b>                        | Keine   |              |                      |
| <b>Prüfung</b>                                | Klausur   |              |                      |
| <b>Prüfungsdauer und -umfang</b>              | 120 min   |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br>Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht<br>Technomathematik: Vertiefung I. Mathematik: Wahlpflicht   |              |                      |

| <b>Lehrveranstaltung L1046: Graphentheorie und Optimierung</b> |   |
|--|---|
| <b>Typ</b>   | Vorlesung   |
| <b>SWS</b>   | 2   |
| <b>LP</b>  | 3   |
| <b>Arbeitsaufwand in Stunden</b>                               | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>  | Prof. Anusch Taraz  |
| <b>Sprachen</b>  | DE/EN   |
| <b>Zeitraum</b>  | SoSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Graphen, Durchlaufen von Graphen, Bäume</li> <li>• Planare Graphen</li> <li>• Kürzeste Wege</li> <li>• Minimale Spannbäume</li> <li>• Maximale Flüsse und minimale Schnitte</li> <li>• Sätze von Menger, König-Egervary, Hall</li> <li>• NP-vollständige Probleme</li> <li>• Backtracking und Heuristiken</li> <li>• Lineare Programmierung</li> <li>• Dualität</li> <li>• Ganzzahlige lineare Programmierung</li> </ul>   |
| <b>Literatur</b>   | <ul style="list-style-type: none"> <li>• M. Aigner: Diskrete Mathematik, Vieweg, 2004</li> <li>• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Algorithmen - Eine Einführung, Oldenbourg, 2013</li> <li>• J. Matousek und J. Nešetřil: Diskrete Mathematik, Springer, 2007</li> <li>• A. Steger: Diskrete Strukturen (Band 1), Springer, 2001</li> <li>• A. Taraz: Diskrete Mathematik, Birkhäuser, 2012</li> <li>• V. Turau: Algorithmische Graphentheorie, Oldenbourg, 2009</li> <li>• K.-H. Zimmermann: Diskrete Mathematik, BoD, 2006</li> </ul> |

| <b>Lehrveranstaltung L1047: Graphentheorie und Optimierung</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                               | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Anusch Taraz                 |
| <b>Sprachen</b>  | DE/EN                              |
| <b>Zeitraum</b>  | SoSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

| <b>Modul M0873: Software-Fachpraktikum</b>    |   |            |           |
|---|---|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |   |            |           |
| <b>Titel</b>                                  | <b>Typ</b>  | <b>SWS</b> | <b>LP</b> |
| <b>Modulverantwortlicher</b>                  | Prof. Karl-Heinz Zimmermann   |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | Grundlagen der Softwaretechnik  |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |            |           |
| <b>Fachkompetenz</b>                          | Die Studierenden kennen wichtige Aspekte und typische Phasen der Software-Projektentwicklung (wie Planung, Spezifikation, Implementierung, Validierung, Wartung und Dokumentation), die in der Praxis mehr oder weniger ausgeprägt zum Tragen kommen.       |            |           |
| <i>Wissen</i>                                 |   |            |           |
| <i>Fertigkeiten</i>                           | Die Studierenden können typische Stufen der Software-Projektentwicklung beschreiben und eng umrissene Teilaufgaben in einem Software-Projekt innerhalb einer Gruppe oder eigenständig bearbeiten.   |            |           |
| <b>Personale Kompetenzen</b>                  | Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben aus der Software-Entwicklung alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.   |            |           |
| <i>Sozialkompetenz</i>                        |   |            |           |
| <i>Selbstständigkeit</i>                      | Die Studierenden sind nach Abschluss des Moduls imstande, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen. |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 180, Präsenzstudium 0  |            |           |
| <b>Leistungspunkte</b>                        | 6   |            |           |
| <b>Studienleistung</b>                        | Keine   |            |           |
| <b>Prüfung</b>                                | Schriftliche Ausarbeitung (laut FPRO)   |            |           |
| <b>Prüfungsdauer und -umfang</b>              | Die Ausarbeitung wird von der Betreuerin bzw. dem Betreuer der Bachelorarbeit bewertet.   |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Kernqualifikation: Pflicht  |            |           |

| <b>Modul M1578: Seminare Informatik</b>       |   |            |            |           |
|---|---|------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |   |            |            |           |
| <b>Titel</b>                                  |   | <b>Typ</b> | <b>SWS</b> | <b>LP</b> |
| Seminar Informatik I (L2362)                  |   | Seminar    | 2          | 3         |
| Seminar Informatik II (L2361)                 |   | Seminar    | 2          | 3         |
| <b>Modulverantwortlicher</b>                  | Prof. Karl-Heinz Zimmermann   |            |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |            |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | Grundlegende Module aus der Informatik und Mathematik auf Bachelorebene.  |            |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |            |            |           |
| <b>Fachkompetenz</b>                          |   |            |            |           |
| <i>Wissen</i>                                 | Die Studierenden können <ul style="list-style-type: none"> <li>• ein spezifisches Thema der Informatik erklären,</li> <li>• komplexe Sachverhalte beschreiben,</li> <li>• unterschiedliche Standpunkte darlegen und kritisch bewerten.</li> </ul>   |            |            |           |
| <i>Fertigkeiten</i>                           | Die Studierenden können <ul style="list-style-type: none"> <li>• sich in einer begrenzten Zeit in ein spezifisches Thema der Informatik einarbeiten,</li> <li>• eine Literaturrecherche durchführen und die Quellen richtig zitieren und angeben,</li> <li>• selbstständig einen Vortrag ausarbeiten und vor ausgewählten Publikum halten,</li> <li>• den Vortrag in einem Abstract zusammenfassen,</li> <li>• im Rahmen der Diskussion Fachfragen beantworten.</li> </ul>              |            |            |           |
| <b>Personale Kompetenzen</b>                  |   |            |            |           |
| <i>Sozialkompetenz</i>                        | Die Studierenden sind in der Lage, <ul style="list-style-type: none"> <li>• ein Thema für eine bestimmte Zielgruppe aufzuarbeiten und darzustellen,</li> <li>• mit der Betreuerin bzw. dem Betreuer das Thema sowie Inhalt und Aufbau des Vortrages zu diskutieren,</li> <li>• einzelne Aspekte aus dem Themengebiet mit den Zuhörerinnen und Zuhörern durchzusprechen,</li> <li>• als Vortragende bzw. Vortragender auf die Fragen der Zuhörerinnen und Zuhörer einzugehen.</li> </ul> |            |            |           |
| <i>Selbstständigkeit</i>                      | Die Studierenden werden die Lage versetzt, <ul style="list-style-type: none"> <li>• eigenständig Aufgaben zu definieren,</li> <li>• notwendiges Wissen zu erschließen,</li> <li>• geeignete Hilfsmittel einzusetzen,</li> <li>• unter Anleitung der Betreuerin bzw. des Betreuers den Arbeitsstand kritisch zu überprüfen.</li> </ul>   |            |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56   |            |            |           |
| <b>Leistungspunkte</b>                        | 6   |            |            |           |
| <b>Studienleistung</b>                        | Keine   |            |            |           |
| <b>Prüfung</b>                                | Referat   |            |            |           |
| <b>Prüfungsdauer und -umfang</b>              | x   |            |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht  |            |            |           |

| <b>Lehrveranstaltung L2362: Seminar Informatik I</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Seminar                            |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>                                      | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>                                      | DE/EN                              |
| <b>Zeitraum</b>                                      | WiSe/SoSe                          |
| <b>Inhalt</b>  |                                    |
| <b>Literatur</b>                                     |                                    |

| <b>Lehrveranstaltung L2361: Seminar Informatik II</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Seminar                            |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>                                       | Prof. Karl-Heinz Zimmermann        |
| <b>Sprachen</b>                                       | DE/EN                              |
| <b>Zeitraum</b>                                       | WiSe/SoSe                          |
| <b>Inhalt</b>   |                                    |
| <b>Literatur</b>                                      |                                    |

| Modul M0672: Signale und Systeme              |  |              |                      |
|---|--|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                    |  |              |                      |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Signale und Systeme (L0432)                   |  | Vorlesung    | 3            4       |
| Signale und Systeme (L0433)                   |  | Gruppenübung | 2            2       |
| <b>Modulverantwortlicher</b>                  | Prof. Gerhard Bauch  |              |                      |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |              |                      |
| <b>Empfohlene Vorkenntnisse</b>               | Mathematik 1-3<br><br>Das Modul führt in das Thema der Signal- und Systemtheorie ein. Sicherer Umgang mit grundlegenden mathematischen Methoden, wie sie in den Modulen Mathematik 1-3 vermittelt werden, wird erwartet. Darüber hinaus sind Vorkenntnisse in Grundlagen von Spektraltransformationen (Fourier-Reihe, Fourier-Transformation, Laplace-Transformation) zwar nützlich, aber keine Voraussetzung.   |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |                      |
| <b>Fachkompetenz</b>                          | Die Studierenden können Signale und lineare zeitinvariante (LTI) Systeme im Sinne der Signal- und Systemtheorie klassifizieren und beschreiben. Sie beherrschen die grundlegenden Integraltransformationen zeitkontinuierlicher und zeitdiskreter deterministischer Signale und Systeme. Sie können deterministische Signale und Systeme in Zeit- und Bildbereich mathematisch beschreiben und analysieren. Sie verstehen elementare Operationen und Konzepte der Signalverarbeitung und können diese in Zeit- und Bildbereich beschreiben. Insbesondere verstehen Sie die mit dem Übergang vom zeitkontinuierlichen zum zeitdiskreten Signal bzw. System einhergehenden Effekte in Zeit- und Bildbereich.   |              |                      |
| <i>Wissen</i>                                 |  |              |                      |
| <i>Fertigkeiten</i>                           | Die Studierenden können deterministische Signale und lineare zeitinvariante Systeme mit den Methoden der Signal- und Systemtheorie beschreiben und analysieren. Sie können einfache Systeme hinsichtlich wichtiger Eigenschaften wie Betrags- und Phasenfrequenzgang, Stabilität, Linearität etc. analysieren und entwerfen. Sie können den Einfluß von LTI-Systemen auf die Signaleigenschaften in Zeit- und Frequenzbereich beurteilen.  |              |                      |
| <b>Personale Kompetenzen</b>                  |  |              |                      |
| <i>Sozialkompetenz</i>                        | Die Studierenden können fachspezifische Aufgaben gemeinsam bearbeiten.   |              |                      |
| <i>Selbstständigkeit</i>                      | Die Studierenden sind in der Lage, die notwendigen Informationen aus geeigneten Literaturquellen selbständig zu beschaffen und in den Kontext der Vorlesung zu setzen. Sie können ihren Wissensstand mit Hilfe vorlesungsbegleitender Maßnahmen (klausurnahe Aufgaben, Software-Tools, Clicker-System) kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern.   |              |                      |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70  |              |                      |
| <b>Leistungspunkte</b>                        | 6  |              |                      |
| <b>Studienleistung</b>                        | Keine  |              |                      |
| <b>Prüfung</b>                                | Klausur  |              |                      |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |              |                      |
| <b>Zuordnung zu folgenden</b>                 | Allgemeine Ingenieurwissenschaften (7 Semester): Kernqualifikation: Pflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Data Science: Kernqualifikation: Pflicht<br>Elektrotechnik: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht |              |                      |

|                  |   |
|------------------|---|
| <b>Curricula</b> | General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Verfahrenstechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Mechatronik: Kernqualifikation: Pflicht<br>Technomathematik: Vertiefung III. Ingenieurwissenschaften: Wahlpflicht |
|------------------|---|

**Lehrveranstaltung L0432: Signale und Systeme**

|                                  |                                    |
|----------------------------------|------------------------------------|
| <b>Typ</b>                       | Vorlesung                          |
| <b>SWS</b>                       | 3                                  |
| <b>LP</b>                        | 4                                  |
| <b>Arbeitsaufwand in Stunden</b> | Eigenstudium 78, Präsenzstudium 42 |
| <b>Dozenten</b>                  | Prof. Gerhard Bauch                |
| <b>Sprachen</b>                  | DE/EN                              |
| <b>Zeitraum</b>                  | SoSe                               |

|               |   |
|---------------|---|
| <b>Inhalt</b> | <ul style="list-style-type: none"> <li>• Einführung in die Signal- und Systemtheorie</li> <li>• Signale                         <ul style="list-style-type: none"> <li>◦ Klassifikation von Signalen                                 <ul style="list-style-type: none"> <li>▪ Zeitkontinuierliche und zeitdiskrete Signale</li> <li>▪ Analoge und digitale Signale</li> <li>▪ Deterministische und zufällige Signale</li> </ul> </li> <li>◦ Beschreibung von LTI-Systemen durch Differentialgleichungen bzw. Differenzgleichungen</li> <li>◦ Grundlegende Eigenschaften von Signalen und grundlegende Operationen</li> <li>◦ Elementare Signale</li> <li>◦ Distributionen</li> <li>◦ Leistung und Energie von Signalen</li> <li>◦ Korrelationsfunktionen deterministischer Signale                                 <ul style="list-style-type: none"> <li>▪ Autokorrelationsfunktion</li> <li>▪ Kreuzkorrelationsfunktion</li> <li>▪ Orthogonale Signale</li> <li>▪ Anwendungen der Korrelation</li> </ul> </li> </ul> </li> <li>• Lineare zeitinvariante Systeme (linear time-invariant (LTI) systems)                         <ul style="list-style-type: none"> <li>◦ Linearität</li> <li>◦ Zeitinvarianz</li> <li>◦ Beschreibung von LTI-Systemen durch Impulsantwort und Übertragungsfunktion</li> <li>◦ Faltung</li> <li>◦ Faltung und Korrelation</li> <li>◦ Eigenschaften von LTI-Systemen</li> <li>◦ Kausale Systeme</li> <li>◦ Stabile Systeme</li> <li>◦ Gedächtnislose Systeme</li> </ul> </li> <li>• Fourier-Reihe und Fourier-Transformation                         <ul style="list-style-type: none"> <li>◦ Fourier-Transformation zeitkontinuierlicher, zeitdiskreter, periodischer und nicht-periodischer Signale</li> <li>◦ Eigenschaften der Fourier-Transformation</li> <li>◦ Fourier-Transformation einiger elementarer Signale</li> <li>◦ Parsevalsches Theorem</li> </ul> </li> <li>• Analyse von LTI-Systemen und Signalen im Frequenzbereich                         <ul style="list-style-type: none"> <li>◦ Übertragungsfunktion, Betragsfrequenzgang, Phasengang</li> <li>◦ Übertragungsfaktor, Dämpfung, Gewinn</li> <li>◦ Frequenzselektive und nicht-frequenzselektive LTI-Systeme</li> <li>◦ Bandbreite-Definitionen</li> <li>◦ Grundlegende Typen von Systemen (Filtern): Tiefpass, Hochpass, Bandpass, Bandsperre</li> <li>◦ Phasenlaufzeit und Gruppenlaufzeit</li> <li>◦ Linearphasige Systeme</li> <li>◦ Verzerrungsfreie Systeme</li> <li>◦ Spektralanalyse mit begrenztem Beobachtungsfenster: Leck-Effekt</li> </ul> </li> <li>• Laplace-Transformation                         <ul style="list-style-type: none"> <li>◦ Zusammenhang von Fourier-Transformation und Laplace-Transformation</li> </ul> </li> </ul> |
|---------------|---|

|                         |  |
|-------------------------|--|
|                         | <ul style="list-style-type: none"> <li>◦ Eigenschaften der Laplace-Transformation</li> <li>◦ Laplace-Transformation einiger elementarer Signale</li> <li>• Analyse von LTI-Systemen im s-Bereich             <ul style="list-style-type: none"> <li>◦ Übertragungsfunktion von LTI-Systemen</li> <li>◦ Zusammenhang von Laplace-Transformation, Betragsfrequenzgang und Phasengang</li> <li>◦ Analyse von LTI-Systemen mit Pol-Nullstellen-Diagrammen</li> <li>◦ Allpass-Filter</li> <li>◦ Minimalphasige, maximalphasige und gemischtphasige Filter</li> <li>◦ Stabile Systeme</li> </ul> </li> <li>• Abtastung             <ul style="list-style-type: none"> <li>◦ Abtasttheorem</li> <li>◦ Rekonstruktion des zeitkontinuierlichen Signals in Frequenz- und Zeitbereich</li> <li>◦ Überabtastung</li> <li>◦ Aliasing</li> <li>◦ Abtastung mit Pulsen endlicher Dauer, Sample and Hold</li> <li>◦ Dezimierung und Interpolation</li> </ul> </li> <li>• Zeitdiskrete Fourier-Transformation (Discrete-Time Fourier Transform (DTFT))             <ul style="list-style-type: none"> <li>◦ Zusammenhang zwischen Fourier-Transformation und DTFT</li> <li>◦ Eigenschaften der DTFT</li> </ul> </li> <li>• Diskrete Fourier-Transformation (Discrete Fourier Transform (DFT))             <ul style="list-style-type: none"> <li>◦ Zusammenhang zwischen DTFT und DFT</li> <li>◦ Zyklische Eigenschaften der DFT</li> <li>◦ DFT-Matrix</li> <li>◦ Zero-Padding</li> <li>◦ Zyklische Faltung</li> <li>◦ Schnelle Fourier-Transformation (Fast Fourier Transform (FFT))</li> <li>◦ Anwendung der DFT: Orthogonal Frequency Division Multiplex (OFDM)</li> </ul> </li> <li>• Z-Transformation             <ul style="list-style-type: none"> <li>◦ Zusammenhang zwischen Laplace-Transformation, DTFT, und z-Transformation</li> <li>◦ Eigenschaften der z-Transformation</li> <li>◦ Z-transform einiger elementarer zeitdiskreter Signale</li> </ul> </li> <li>• Zeitdiskrete Systeme, Digitale Filter             <ul style="list-style-type: none"> <li>◦ FIR und IIR Filter</li> <li>◦ Z-Transformation digitaler Filter</li> <li>◦ Analyse zeitdiskreter Systeme mit Pol-Nullstellen-Diagrammen im z-Bereich</li> <li>◦ Stabilität</li> <li>◦ Allpass-Filter</li> <li>◦ Minimalphasige, maximalphasige und gemischtphasige Filter</li> <li>◦ Linearphasige Filter</li> </ul> </li> </ul> |
| <p><b>Literatur</b></p> | <ul style="list-style-type: none"> <li>• T. Frey , M. Bossert , Signal- und Systemtheorie, B.G. Teubner Verlag 2004</li> <li>• K. Kammeyer, K. Kroschel, Digitale Signalverarbeitung, Teubner Verlag.</li> <li>• B. Girod ,R. Rabensteiner , A. Stenger , Einführung in die Systemtheorie, B.G. Teubner, Stuttgart, 1997</li> <li>• J.R. Ohm, H.D. Lüke , Signalübertragung, Springer-Verlag 8. Auflage, 2002</li> <li>• S. Haykin, B. van Veen: Signals and systems. Wiley.</li> <li>• Oppenheim, A.S. Willsky: Signals and Systems. Pearson.</li> <li>• Oppenheim, R. W. Schafer: Discrete-time signal processing. Pearson.</li> </ul>   |

| <b>Lehrveranstaltung L0433: Signale und Systeme</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>                                     | Prof. Gerhard Bauch                |
| <b>Sprachen</b>                                     | DE/EN                              |
| <b>Zeitraum</b>                                     | SoSe                               |
| <b>Inhalt</b>                                       | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                    | Siehe korrespondierende Vorlesung  |

## Fachmodule der Vertiefung I. Computer- und Software-Engineering

In dieser Vertiefung finden sich die Gebiete Softwaretechnik und Technische Informatik.

Die Softwaretechnik beschäftigt sich mit Planung, Entwurf, Analyse, Betrieb und Dokumentation von Softwaresystemen sowie der Spezifikation der dazu notwendigen Datenstrukturen. Die Vertiefung bietet einerseits Module aus dem Kernbereich der Softwaretechnik, wie Compilerbau, Datenbanken und Software-Entwicklung, und andererseits anwendungsorientierte Module, wie Verteilte Systeme, Kombinatorische Strukturen und Algorithmen sowie Rechnergestützte Geometrie.

Die Technische Informatik befasst sich mit Aufbau, Entwurf, Organisation, Bewertung und Betrieb von Rechnersystemen. Sie ist Schnittstelle zwischen Informatik und Elektrotechnik und beschreibt Rechnersysteme hauptsächlich aus elektrologischer Sicht. Kenntnisse aus den grundständigen Veranstaltungen über Elektrotechnik werden nicht vorausgesetzt. Die Vertiefung bietet zum einen Module aus dem Kernbereich der Technischen Informatik, wie Recherarchitektur, Eingebettete Systeme sowie ein entsprechendes Labor-Praktikum, und andererseits anwendungsorientierte Module, wie Medizintechnische Systeme und die klassische Nachrichtentechnik. Die Einführungsveranstaltung über Quantenmechanik spielt eine fundamentale Rolle beim Aufbau von Quantenrechnern.

| <b>Modul M0625: Databases</b>                 |  |
|---|--|
| <b>Lehrveranstaltungen</b>                    |  |
| <b>Titel</b>                                  | <b>Typ</b> <b>SWS</b> <b>LP</b>  |
| Datenbanken (L0337)                           | Vorlesung                      4                      5  |
| Datenbanken (L1150)                           | Projekt-/problembasierte<br>Lehrveranstaltung                      1                      1  |
| <b>Modulverantwortlicher</b>                  | NN   |
| <b>Zulassungsvoraussetzungen</b>              | None   |
| <b>Empfohlene Vorkenntnisse</b>               | Students should have basic knowledge in the following areas: <ul style="list-style-type: none"> <li>• Discrete Algebraic Structures</li> <li>• Procedural Programming</li> <li>• Logic, Automata, and Formal Languages</li> <li>• Object-Oriented Programming, Algorithms and Data Structures</li> </ul>   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                          | Students can explain the general architecture of an application system that is based on a database. They describe the syntax and semantics of the Entity Relationship conceptual modeling languages, and they can enumerate basic decision problems and know which features of a domain model can be captured with ER and which features cannot be represented. Furthermore, students can summarize the features of the relational data model, and can describe how ER models can be systematically transformed into the relational data model. Student are able to discuss dependency theory using the operators of relational algebra, and they know how to use relational algebra as a query language. In addition, they can sketch the main modules of the architecture of a database system from an implementation point of view. Storage and index structures as well as query answering and optimization techniques can be explained. The role of transactions can be described in terms of ACID conditions and common recovery mechanisms can be characterized. The students can recall why recursion is important for query languages and describe how Datalog can be used and implemented. They demonstrate how Datalog can be used for information integration. For solving ER decision problems the students can explain description logics with their syntax and semantics, they describe description logic decision problems and explain how these problems can be mapped onto each other. They can sketch the idea of ontology-based data access and can name the main complexity measure in database theory. Last but not least, the students can describe the main features of XML and can explain XPath and XQuery as query languages. |
| <i>Wissen</i>                                 | Students can apply ER for describing domains for which they receive a textual  |

|  |   |
|--|---|
| <p><i>Fertigkeiten</i></p> <p><b>Personale Kompetenzen</b></p> <p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i></p> | <p>description, and students can transform relational schemata with a given set of functional dependencies into third normal form or even Boyce-Codd normal form. They can also apply relational algebra, SQL, or Datalog to specify queries. Using specific datasets, they can explain how index structures work (e.g., B-trees) and how index structures change while data is added or deleted. They can rewrite queries for better performance of query evaluation. Students can analyse which query language expressivity is required for which application problem. Description logics can be applied for domain modeling, and students can transform ER diagrams into description logics in order to check for consistency and implicit subsumption relations. They solve data integration problems using Datalog and LAV or GAV rules. Students can apply XPath and Xquery to retrieve certain patterns in XML data.</p> <p>Students develop an understanding of social structures in a company used for developing real-world products. They know the responsibilities of data analysts, programmers, and managers in the overall production process.</p> |
| <b>Arbeitsaufwand in Stunden</b>   | Eigenstudium 110, Präsenzstudium 70   |
| <b>Leistungspunkte</b>   | 6   |
| <b>Studienleistung</b>   | Keine   |
| <b>Prüfung</b>   | Klausur   |
| <b>Prüfungsdauer und -umfang</b>   | 90 min  |
| <b>Zuordnung zu folgenden Curricula</b>  | <p>Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht</p> <p>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht</p> <p>Data Science: Kernqualifikation: Pflicht</p> <p>Technomathematik: Vertiefung II. Informatik: Wahlpflicht</p>  |

| <b>Lehrveranstaltung L0337: Databases</b> |   |
|---|---|
| <b>Typ</b>                                | Vorlesung   |
| <b>SWS</b>                                | 4   |
| <b>LP</b>                                 | 5   |
| <b>Arbeitsaufwand in Stunden</b>          | Eigenstudium 94, Präsenzstudium 56  |
| <b>Dozenten</b>                           | NN  |
| <b>Sprachen</b>                           | EN  |
| <b>Zeitraum</b>                           | WiSe  |
| <b>Inhalt</b>                             | <ul style="list-style-type: none"> <li>• Architecture of database systems, conceptual data modeling with the Entity Relationship (ER) modeling language</li> <li>• Relational data model, referential integrity, keys, foreign keys, functional dependencies (FDs), canonical mapping of entity types and relationship into the relational data model, anomalies</li> <li>• Relational algebra as a simple query language</li> <li>• Dependency theory, FD closure, canonical cover of FD set, decomposition of relational schemata, multivalued dependencies, normalization, inclusion dependencies</li> <li>• Practical query languages and integrity constraints w/o considering a conceptual domain model: SQL</li> <li>• Storage structures, database implementation architecture</li> <li>• Index structures</li> <li>• Query processing</li> <li>• Query optimization</li> <li>• Transactions and recovery</li> <li>• Query languages with recursion and consideration of a simple conceptual domain model: Datalog</li> <li>• Semi-naive evaluation strategy, magic sets transformation</li> <li>• Information integration, declarative schema transformation (LAV, GAV), distributed database systems</li> <li>• Description logics, syntax, semantics, decision problems, decision algorithms for Abox satisfiability</li> <li>• Ontology based data access (OBDA), DL-Lite for formalizing ER diagrams</li> <li>• Complexity measure: Data complexity</li> <li>• Semistructured databases and query languages: XML and XQuery</li> </ul> |
| <b>Literatur</b>                          | <ol style="list-style-type: none"> <li>1. A. Kemper, A. Eickler, Datenbanksysteme - n. Auflage, Oldenbourg, 2010</li> <li>2. S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995</li> <li>3. Database Systems, An Application Oriented Approach, Pearson International Edition, 2005</li> <li>4. H. Garcia-Molina, J.D. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002</li> </ol>   |

| <b>Lehrveranstaltung L1150: Databases</b> |  |
|---|--|
| <b>Typ</b>                                | Projekt-/problembasierte Lehrveranstaltung |
| <b>SWS</b>                                | 1  |
| <b>LP</b>                                 | 1  |
| <b>Arbeitsaufwand in Stunden</b>          | Eigenstudium 16, Präsenzstudium 14         |
| <b>Dozenten</b>                           | NN   |
| <b>Sprachen</b>                           | EN   |
| <b>Zeitraum</b>                           | WiSe                                       |
| <b>Inhalt</b>                             | Siehe korrespondierende Vorlesung          |
| <b>Literatur</b>                          | Siehe korrespondierende Vorlesung          |

| Modul M1586: Wissenschaftliche Programmierung |  |              |                      |
|---|--|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                    |  |              |                      |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Wissenschaftliche Programmierung (L2405)      |  | Vorlesung    | 3            4       |
| Wissenschaftliche Programmierung (L2406)      |  | Gruppenübung | 2            2       |
| <b>Modulverantwortlicher</b>                  | Prof. Tobias Knopp   |              |                      |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |              |                      |
| <b>Empfohlene Vorkenntnisse</b>               | Prozedurale Programmierung, Lineare Algebra  |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |                      |
| <b>Fachkompetenz</b>                          | Die Studierenden   |              |                      |
| <i>Wissen</i>                                 | <ul style="list-style-type: none"> <li>• können wissenschaftliche Probleme in einer modernen Programmiersprache effizient lösen.</li> <li>• sind mit dem Konzept der reproduzierbaren Wissenschaft vertraut.</li> <li>• können mit mehrdimensionalen Arrays, sparse Arrays, Data Frames (tabellenförmige Daten) und Missing Data umgehen. Sie kennen sie Vor- und Nachteile spezifischer Datenstrukturen.</li> <li>• kennen verschiedene Möglichkeiten um Daten, Datenbeziehungen und Fehlermaße geeignet darzustellen. Sie kennen bekannte Datenformate zur Speicherung von wissenschaftlichen Daten und können für spezifische Daten ein geeignetes Format auswählen.</li> </ul>   |              |                      |
| <i>Fertigkeiten</i>                           | <p>Sie sind in der Lage</p> <ul style="list-style-type: none"> <li>• komplexe Probleme aus einer mathematischen Formulierung in eine geeignetes Programm zu übersetzen.</li> <li>• ein komplexes Problem in Teilprobleme aufzuteilen welche modular umgesetzt werden können.</li> <li>• numerische Standardprobleme zu identifizieren und hierfür geeignete Standardalgorithmen nutzen, die in Bibliotheken vorhanden sind.</li> <li>• wartbaren Programmcode zu schreiben, dessen Korrektheit durch geeignete Tests überprüft wird.</li> <li>• die Laufzeit von Programmen zu messen, Flaschenhalse zu identifizieren und geeignete Beschleunigungstechniken anzuwenden.</li> </ul> |              |                      |
| <b>Personale Kompetenzen</b>                  | Die Studierenden können in sowohl selbstständig als auch in Teams an komplexen Problemen arbeiten. Sie können sich untereinander austauschen und ihre individuellen Stärken zur Lösung des Problems einbringen.  |              |                      |
| <i>Sozialkompetenz</i>                        | Die Studierenden sind in der Lage ein komplexes Problem eigenständig zu untersuchen und einzuschätzen, welche Kompetenzen zur Lösung des Problems benötigt werden.   |              |                      |
| <i>Selbstständigkeit</i>                      |  |              |                      |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70  |              |                      |
| <b>Leistungspunkte</b>                        | 6  |              |                      |
| <b>Studienleistung</b>                        | Keine  |              |                      |
| <b>Prüfung</b>                                | Klausur  |              |                      |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>Data Science: Kernqualifikation: Pflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht  |              |                      |

| <b>Lehrveranstaltung L2405: Wissenschaftliche Programmierung</b> |  |
|--|--|
| <b>Typ</b>   | Vorlesung  |
| <b>SWS</b>   | 3  |
| <b>LP</b>  | 4  |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 78, Präsenzstudium 42   |
| <b>Dozenten</b>  | Prof. Tobias Knopp   |
| <b>Sprachen</b>  | DE   |
| <b>Zeitraum</b>  | SoSe   |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Elementare Datentypen und der Zusammenhang zur Mathematik</li> <li>• Wissenschaftliche Datentypen: Mehrdimensionale Arrays, sparse Arrays, Data Frames, Missing Data</li> <li>• Multiple Dispatch als effizientes Paradigma für die wissenschaftliche Programmierung</li> <li>• Literate Programming</li> <li>• Profiling und Benchmarks</li> <li>• Beschleunigungstechniken: Caching, Multi-threading, SIMD, GPGPU</li> <li>• Wissenschaftliche Datenformate: CSV, TOML, HDF5, und ausgewählte Beispiele</li> <li>• Datenvisualisierung</li> <li>• Numerische Standardtechniken und effiziente Programmbibliotheken (BLAS, LAPACK, FFTW, ...)</li> <li>• Tests, Codeverwaltung, Dokumentation</li> <li>• Reproduzierbare Wissenschaft</li> </ul> |
| <b>Literatur</b>   | Ben Lauwens, Allen Downey: Think Julia: How to Think Like a Computer Scientist   |

| <b>Lehrveranstaltung L2406: Wissenschaftliche Programmierung</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Tobias Knopp                 |
| <b>Sprachen</b>  | DE                                 |
| <b>Zeitraum</b>  | SoSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

| Modul M0971: Betriebssysteme                  |  |            |           |
|---|--|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |            |           |
| <b>Titel</b>                                  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Betriebssysteme (L1153)                       | Vorlesung  | 2          | 3         |
| Betriebssysteme (L1154)                       | Gruppenübung   | 2          | 3         |
| <b>Modulverantwortlicher</b>                  | Prof. Volker Turau   |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Prozedurales Programmieren</li> <li>• Objekt-orientierte Programmierung, Algorithmen und Datenstrukturen</li> <li>• Erfahrung in der Anwendung von betriebssystemnahen Werkzeugen wie Editoren, Linker, Compiler</li> <li>• Erfahrung im Umgang mit C-Bibilotheken</li> </ul>   |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |            |           |
| <b>Fachkompetenz</b>                          | <p><i>Wissen</i></p> <p>Studierende können die wichtigsten Abstraktion von Betriebssystem erklären (Prozess, virtueller Speicher, Datei, Deadlock, Livelock). Sie sind in der Lage, die Prozesszustände und die dazugehörigen Übergänge zu beschreiben. Sie kennen die wichtigsten Architekturvarianten von Betriebssystemen und können existierende Betriebssysteme diesen Varianten zuordnen. Die Teilnehmer sind in der Lage, nebenläufige Programm mittels Threads, conditional Variablen und Semaphoren zu erstellen. Sie können mehrere Varianten zur Realisierung von Filesystemen erläutern. Des Weiteren können sie mindestens drei Scheduling Algorithmen erläutern.</p> <p><i>Fertigkeiten</i></p> <p>Studierende können die POSIX Bibliotheken zur nebenläufigen Programmierung korrekt und effizient einsetzen. Sie sind in der Lage für eine Scheduling Aufgabe unter gegebenen Randbedingungen die Effeizienz eines Scheduling-Algorithmus zu beurteilen.</p> |            |           |
| <b>Personale Kompetenzen</b>                  | <p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i></p>  |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |            |           |
| <b>Leistungspunkte</b>                        | 6  |            |           |
| <b>Studienleistung</b>                        | Keine  |            |           |
| <b>Prüfung</b>                                | Klausur  |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 90 min   |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Computer Science: Kernqualifikation: Pflicht<br>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht   |            |           |

| <b>Lehrveranstaltung L1153: Betriebssysteme</b> |   |
|---|---|
| <b>Typ</b>                                      | Vorlesung   |
| <b>SWS</b>                                      | 2   |
| <b>LP</b>                                       | 3   |
| <b>Arbeitsaufwand in Stunden</b>                | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>                                 | Prof. Volker Turau  |
| <b>Sprachen</b>                                 | DE  |
| <b>Zeitraum</b>                                 | SoSe  |
| <b>Inhalt</b>                                   | <ul style="list-style-type: none"> <li>• Architekturen für Betriebssysteme</li> <li>• Prozesse</li> <li>• Nebenläufigkeit</li> <li>• Verklemmungen</li> <li>• Speicherverwaltung</li> <li>• Scheduling</li> <li>• Dateisysteme</li> </ul> |
| <b>Literatur</b>                                | <ol style="list-style-type: none"> <li>1. Operating Systems, William Stallings, Pearson International Edition</li> <li>2. Moderne Betriebssysteme, Andrew Tanenbaum, Pearson Studium</li> </ol>   |

| <b>Lehrveranstaltung L1154: Betriebssysteme</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>                                      | Gruppenübung                       |
| <b>SWS</b>                                      | 2                                  |
| <b>LP</b>                                       | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>                                 | Prof. Volker Turau                 |
| <b>Sprachen</b>                                 | DE                                 |
| <b>Zeitraum</b>                                 | SoSe                               |
| <b>Inhalt</b>                                   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                | Siehe korrespondierende Vorlesung  |

| Modul M0791: Rechnerarchitektur               |   |   |   |
|---|---|---|---|
| <b>Lehrveranstaltungen</b>                    |   |   |   |
| <b>Titel</b>                                  |   | <b>Typ</b>                                    | <b>SWS</b> <b>LP</b>                                  |
| Rechnerarchitektur (L0793)                    |   | Vorlesung                                     | 2            3  |
| Rechnerarchitektur (L0794)                    |   | Projekt-/problembasierte<br>Lehrveranstaltung | 2            2  |
| Rechnerarchitektur (L1864)                    |   | Gruppenübung                                  | 1            1  |
| <b>Modulverantwortlicher</b>                  | Prof. Heiko Falk  |   |   |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |   |   |
| <b>Empfohlene Vorkenntnisse</b>               | Modul "Technische Informatik"   |   |   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |   |   |
| <b>Fachkompetenz</b>                          | <p>In diesem Modul werden fortgeschrittene Konzepte der Rechnerarchitektur vorgestellt. Am Anfang steht ein breiter Überblick über mögliche Programmiermodelle, wie sie für Universalrechner aber auch für spezielle Maschinen (z.B. Signalprozessoren) entwickelt wurden. Anschließend werden prinzipielle Aspekte der Mikroarchitektur von Prozessoren behandelt. Der Schwerpunkt liegt hierbei insbesondere auf dem sogenannten Pipelining und den in diesem Zusammenhang angewandten Methoden zur Beschleunigung der Befehlsausführung. Die Studierenden lernen Mechanismen zum dynamischen Scheduling, zur Sprungvorhersage, zu superskalaren Architekturen und zu Speicher-Hierarchien kennen.</p>  |   |   |
| <i>Wissen</i>                                 | <p>Die Studierenden sind in der Lage, den Aufbau eines Prozessors zu erklären. Sie kennen die verschiedenen Architekturprinzipien und Programmiermodelle. Die Studierenden untersuchen verschiedene Strukturen von Pipeline-Architekturen und sind in der Lage, deren Konzepte zu erklären und im Hinblick auf Kriterien wie Performance und Energieeffizienz zu analysieren. Sie bewerten unterschiedliche Speicherarchitekturen, kennen parallele Rechnerarchitekturen und können zwischen Befehls- und Datenparallelität unterscheiden.</p>  |   |   |
| <i>Fertigkeiten</i>                           | <p>Die Studierenden sind nach Abschluss des Moduls in der Lage, ähnliche Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.</p>  |   |   |
| <b>Personale Kompetenzen</b>                  | <p>Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.</p>  |   |   |
| <i>Sozialkompetenz</i>                        |   |   |   |
| <i>Selbstständigkeit</i>                      |   |   |   |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 110, Präsenzstudium 70   |   |   |
| <b>Leistungspunkte</b>                        | 6   |   |   |
| <b>Studienleistung</b>                        | <b>Verpflichtend</b>  | <b>Anteil</b>                                 | <b>Art der Studienleistung</b> <b>Beschreibung</b>    |
|   | Nein  | 15 %  | Fachtheoretisch-<br>fachpraktische<br>Studienleistung |
| <b>Prüfung</b>                                | Klausur   |   |   |
| <b>Prüfungsdauer und -umfang</b>              | 90 Min., Vorlesungsstoff + 4 Testate zur PBL "Rechnerarchitektur"   |   |   |
| <b>Zuordnung zu folgenden Curricula</b>       | <p>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br/>                     Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br/>                     Flugzeug-Systemtechnik: Vertiefung Avionik und eingebettete Systeme: Wahlpflicht<br/>                     General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br/>                     Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br/>                     Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht</p> |   |   |

| <b>Lehrveranstaltung L0793: Rechnerarchitektur</b> |  |
|--|--|
| <b>Typ</b>   | Vorlesung  |
| <b>SWS</b>   | 2  |
| <b>LP</b>  | 3  |
| <b>Arbeitsaufwand in Stunden</b>                   | Eigenstudium 62, Präsenzstudium 28   |
| <b>Dozenten</b>                                    | Prof. Heiko Falk   |
| <b>Sprachen</b>                                    | DE/EN  |
| <b>Zeitraum</b>                                    | WiSe   |
| <b>Inhalt</b>                                      | <ul style="list-style-type: none"> <li>• Einführung</li> <li>• Grundlagen von VHDL</li> <li>• Programmiermodelle</li> <li>• Realisierung elementarer Datentypen</li> <li>• Dynamisches Scheduling</li> <li>• Sprungvorhersage</li> <li>• Superskalare Maschinen</li> <li>• Speicher-Hierarchien</li> </ul> <p>Die Gruppenübungen vertiefen die Vorlesungsinhalte durch Bearbeiten und Besprechen von Übungsblättern und dienen somit zur Klausur-Vorbereitung. Der praktische Umgang mit Fragestellungen aus der Rechnerarchitektur wird in der FPGA-basierten PBL zur Rechnerarchitektur vermittelt, deren Teilnahme verpflichtend ist.</p> |
| <b>Literatur</b>                                   | <ul style="list-style-type: none"> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> </ul>  |

| <b>Lehrveranstaltung L0794: Rechnerarchitektur</b> |  |
|--|--|
| <b>Typ</b>   | Projekt-/problembasierte Lehrveranstaltung |
| <b>SWS</b>   | 2  |
| <b>LP</b>  | 2  |
| <b>Arbeitsaufwand in Stunden</b>                   | Eigenstudium 32, Präsenzstudium 28         |
| <b>Dozenten</b>                                    | Prof. Heiko Falk                           |
| <b>Sprachen</b>                                    | DE/EN                                      |
| <b>Zeitraum</b>                                    | WiSe                                       |
| <b>Inhalt</b>                                      | Siehe korrespondierende Vorlesung          |
| <b>Literatur</b>                                   | Siehe korrespondierende Vorlesung          |

| <b>Lehrveranstaltung L1864: Rechnerarchitektur</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 1                                  |
| <b>LP</b>  | 1                                  |
| <b>Arbeitsaufwand in Stunden</b>                   | Eigenstudium 16, Präsenzstudium 14 |
| <b>Dozenten</b>                                    | Prof. Heiko Falk                   |
| <b>Sprachen</b>                                    | DE/EN                              |
| <b>Zeitraum</b>                                    | WiSe                               |
| <b>Inhalt</b>                                      | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                   | Siehe korrespondierende Vorlesung  |

| Modul M0972: Verteilte Systeme                |   |            |           |
|---|---|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |   |            |           |
| <b>Titel</b>                                  | <b>Typ</b>  | <b>SWS</b> | <b>LP</b> |
| Verteilte Systeme (L1155)                     | Vorlesung   | 2          | 3         |
| Verteilte Systeme (L1156)                     | Gruppenübung  | 2          | 3         |
| <b>Modulverantwortlicher</b>                  | Prof. Volker Turau  |            |           |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Prozedurales Programmieren</li> <li>• Objektorientiertes Programmieren mit Java</li> <li>• Rechnernetze</li> <li>• Socket Programmierung</li> </ul>  |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |            |           |
| <b>Fachkompetenz</b>                          | <p><i>Wissen</i></p> <p>Studierende können die wichtigsten Abstraktion von Verteilten Systemen erklären (Marshalling, Proxy, Dienst, Adresse, Entfernter Aufruf, synchrones/asynchrones System). Sie sind in der Lage, die Vor- und Nachteile verschiedener Arten von Interprozesskommunikation zu beschreiben. Sie kennen die wichtigsten Architekturvarianten von Verteilten Systemen einschließlich ihrer Vor- und Nachteile. Die Teilnehmer sind in der Lage, mindestens drei Synchronisationsverfahren zu beschreiben.</p> <p><i>Fertigkeiten</i></p> <p>Studierende können auf unterschiedliche Arten verteilte Systeme realisieren. Dabei können sie folgende Methoden verwenden:</p> <ul style="list-style-type: none"> <li>• Eigenes Protokoll entwerfen und mittels TCP umsetzen</li> <li>• HTTP als entfernten Aufruf nutzen</li> <li>• RMI als Middleware nutzen</li> </ul> |            |           |
| <b>Personale Kompetenzen</b>                  | <p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i></p>   |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56   |            |           |
| <b>Leistungspunkte</b>                        | 6   |            |           |
| <b>Studienleistung</b>                        | Keine   |            |           |
| <b>Prüfung</b>                                | Klausur   |            |           |
| <b>Prüfungsdauer und -umfang</b>              | 120 min   |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht  |            |           |

| <b>Lehrveranstaltung L1155: Verteilte Systeme</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>                  | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>                                   | Prof. Volker Turau  |
| <b>Sprachen</b>                                   | DE  |
| <b>Zeitraum</b>                                   | WiSe  |
| <b>Inhalt</b>                                     | <ul style="list-style-type: none"> <li>• Architekturen für verteilte Systeme</li> <li>• HTTP: Einfacher entfernter Aufruf</li> <li>• Client-Server Architekturen</li> <li>• Entfernter Aufruf</li> <li>• Remote Method Invocation (RMI)</li> <li>• Synchronisierung</li> <li>• Verteiltes Caching</li> <li>• Namensdienste</li> <li>• Verteilte Dateisysteme</li> </ul> |
| <b>Literatur</b>                                  | <ul style="list-style-type: none"> <li>• Verteilte Systeme - Prinzipien und Paradigmen, Andrew S. Tanenbaum, Maarten van Steen, Pearson Studium</li> <li>• Verteilte Systeme, G. Coulouris, J. Dollimore, T. Kindberg, 2005, Pearson Studium</li> </ul>   |

| <b>Lehrveranstaltung L1156: Verteilte Systeme</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                  | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>                                   | Prof. Volker Turau                 |
| <b>Sprachen</b>                                   | DE                                 |
| <b>Zeitraum</b>                                   | WiSe                               |
| <b>Inhalt</b>                                     | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                  | Siehe korrespondierende Vorlesung  |

| Modul M0953: Introduction to Information Security |  |              |                      |
|---|--|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                        |  |              |                      |
| <b>Titel</b>                                      |  | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Einführung in die Informationssicherheit (L1114)  |  | Vorlesung    | 3            3       |
| Einführung in die Informationssicherheit (L1115)  |  | Gruppenübung | 2            3       |
| <b>Modulverantwortlicher</b>                      | Prof. Dieter Gollmann  |              |                      |
| <b>Zulassungsvoraussetzungen</b>                  | None   |              |                      |
| <b>Empfohlene Vorkenntnisse</b>                   | Basics of Computer Science   |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b>     | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |                      |
| <b>Fachkompetenz</b>                              | Students can   |              |                      |
| <i>Wissen</i>                                     | <ul style="list-style-type: none"> <li>• name the main security risks when using Information and Communication Systems and name the fundamental security mechanisms,</li> <li>• describe commonly used methods for risk and security analysis,</li> <li>• name the fundamental principles of data protection.</li> </ul> |              |                      |
| <i>Fertigkeiten</i>                               | Students can <ul style="list-style-type: none"> <li>• evaluate the strenghts and weaknesses of the fundamental security mechanisms and of the commonly used methods for risk and security analysis,</li> <li>• apply the fundamental principles of data protection to concrete cases.</li> </ul>                         |              |                      |
| <b>Personale Kompetenzen</b>                      |  |              |                      |
| <i>Sozialkompetenz</i>                            | Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.   |              |                      |
| <i>Selbstständigkeit</i>                          | None   |              |                      |
| <b>Arbeitsaufwand in Stunden</b>                  | Eigenstudium 110, Präsenzstudium 70  |              |                      |
| <b>Leistungspunkte</b>                            | 6  |              |                      |
| <b>Studienleistung</b>                            | Keine  |              |                      |
| <b>Prüfung</b>                                    | Klausur  |              |                      |
| <b>Prüfungsdauer und -umfang</b>                  | 120 Minuten  |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>           | Computer Science: Kernqualifikation: Pflicht<br>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>Data Science: Kernqualifikation: Pflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht   |              |                      |

| <b>Lehrveranstaltung L1114: Introduction to Information Security</b> |   |
|--|---|
| <b>Typ</b>   | Vorlesung   |
| <b>SWS</b>   | 3   |
| <b>LP</b>  | 3   |
| <b>Arbeitsaufwand in Stunden</b>                                     | Eigenstudium 48, Präsenzstudium 42  |
| <b>Dozenten</b>  | Prof. Dieter Gollmann   |
| <b>Sprachen</b>  | EN  |
| <b>Zeitraum</b>  | WiSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Fundamental concepts</li> <li>• Passwords &amp; biometrics</li> <li>• Introduction to cryptography</li> <li>• Sessions, SSL/TLS</li> <li>• Certificates, electronic signatures</li> <li>• Public key infrastructures</li> <li>• Side-channel analysis</li> <li>• Access control</li> <li>• Privacy</li> <li>• Software security basics</li> <li>• Security management &amp; risk analysis</li> <li>• Security evaluation: Common Criteria</li> </ul> |
| <b>Literatur</b>   | <p>D. Gollmann: Computer Security, Wiley &amp; Sons, third edition, 2011</p> <p>Ross Anderson: Security Engineering, Wiley &amp; Sons, second edition, 2008</p>   |

| <b>Lehrveranstaltung L1115: Introduction to Information Security</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>                                     | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>  | Prof. Dieter Gollmann              |
| <b>Sprachen</b>  | EN                                 |
| <b>Zeitraum</b>  | WiSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

| <b>Modul M0754: Compiler Construction</b>     |  |              |                      |
|---|--|--------------|----------------------|
| <b>Lehrveranstaltungen</b>                    |  |              |                      |
| <b>Titel</b>                                  |  | <b>Typ</b>   | <b>SWS</b> <b>LP</b> |
| Compilerbau (L0703)                           |  | Vorlesung    | 2            2       |
| Compilerbau (L0704)                           |  | Gruppenübung | 2            4       |
| <b>Modulverantwortlicher</b>                  | Prof. Sibylle Schupp   |              |                      |
| <b>Zulassungsvoraussetzungen</b>              | None   |              |                      |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Practical programming experience</li> <li>• Automata theory and formal languages</li> <li>• Functional programming or procedural programming</li> <li>• Object-oriented programming, algorithms, and data structures</li> <li>• Basic knowledge of software engineering</li> </ul>  |              |                      |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |                      |
| <b>Fachkompetenz</b>                          | <p>Students explain the workings of a compiler and break down a compilation task in different phases. They apply and modify the major algorithms for compiler construction and code improvement. They can re-write those algorithms in a programming language, run and test them. They choose appropriate internal languages and representations and justify their choice. They explain and modify implementations of existing compiler frameworks and experiment with frameworks and tools.</p> <p>Students design and implement arbitrary compilation phases. They integrate their code in existing compiler frameworks. They organize their compiler code properly as a software project. They generalize algorithms for compiler construction to algorithms that analyze or synthesize software.</p> <p>Students develop the software in a team. They explain problems and solutions to their team members. They present and defend their software in class. They communicate in English.</p> <p>Students develop their software independently and define milestones by themselves. They receive feedback throughout the entire project. They organize the software project so that they can assess their progress themselves.</p> |              |                      |
| <i>Wissen</i>                                 |  |              |                      |
| <i>Fertigkeiten</i>                           |  |              |                      |
| <b>Personale Kompetenzen</b>                  |  |              |                      |
| <i>Sozialkompetenz</i>                        |  |              |                      |
| <i>Selbstständigkeit</i>                      |  |              |                      |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |              |                      |
| <b>Leistungspunkte</b>                        | 6  |              |                      |
| <b>Studienleistung</b>                        | Keine  |              |                      |
| <b>Prüfung</b>                                | Fachtheoretisch-fachpraktische Arbeit  |              |                      |
| <b>Prüfungsdauer und -umfang</b>              | Software (Compiler)  |              |                      |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht   |              |                      |

| <b>Lehrveranstaltung L0703: Compiler Construction</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 2   |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 32, Präsenzstudium 28  |
| <b>Dozenten</b>                                       | Prof. Sibylle Schupp  |
| <b>Sprachen</b>                                       | EN  |
| <b>Zeitraum</b>                                       | SoSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Lexical and syntactic analysis</li> <li>• Semantic analysis</li> <li>• High-level optimization</li> <li>• Intermediate languages and code generation</li> <li>• Compilation pipeline</li> </ul>  |
| <b>Literatur</b>                                      | <p>Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam, Compilers: Principles, Techniques, and Tools, 2nd edition</p> <p>Aarne Ranta, Implementing Programming Languages, An Introduction to Compilers and Interpreters, with an appendix coauthored by Markus Forsberg, College Publications, London, 2012</p> |

| <b>Lehrveranstaltung L0704: Compiler Construction</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 4                                  |
| <b>Arbeitsaufwand in Stunden</b>                      | Eigenstudium 92, Präsenzstudium 28 |
| <b>Dozenten</b>                                       | Prof. Sibylle Schupp               |
| <b>Sprachen</b>                                       | EN                                 |
| <b>Zeitraum</b>                                       | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                      | Siehe korrespondierende Vorlesung  |

| Modul M0803: Embedded Systems                 |  |               |  |                     |
|---|--|---------------|--|---------------------|
| <b>Lehrveranstaltungen</b>                    |  |               |  |                     |
| <b>Titel</b>                                  |  | <b>Typ</b>    | <b>SWS</b>                                     | <b>LP</b>           |
| Eingebettete Systeme (L0805)                  |  | Vorlesung     | 3  | 4                   |
| Eingebettete Systeme (L0806)                  |  | Gruppenübung  | 1  | 2                   |
| <b>Modulverantwortlicher</b>                  | Prof. Heiko Falk   |               |  |                     |
| <b>Zulassungsvoraussetzungen</b>              | None   |               |  |                     |
| <b>Empfohlene Vorkenntnisse</b>               | Computer Engineering   |               |  |                     |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |               |  |                     |
| <b>Fachkompetenz</b>                          | <p>Embedded systems can be defined as information processing systems embedded into enclosing products. This course teaches the foundations of such systems. In particular, it deals with an introduction into these systems (notions, common characteristics) and their specification languages (models of computation, hierarchical automata, specification of distributed systems, task graphs, specification of real-time applications, translations between different models).</p>   |               |  |                     |
| <i>Wissen</i>                                 | <p>Another part covers the hardware of embedded systems: Sensors, A/D and D/A converters, real-time capable communication hardware, embedded processors, memories, energy dissipation, reconfigurable logic and actuators. The course also features an introduction into real-time operating systems, middleware and real-time scheduling. Finally, the implementation of embedded systems using hardware/software co-design (hardware/software partitioning, high-level transformations of specifications, energy-efficient realizations, compilers for embedded processors) is covered.</p>  |               |  |                     |
| <i>Fertigkeiten</i>                           | <p>After having attended the course, students shall be able to realize simple embedded systems. The students shall realize which relevant parts of technological competences to use in order to obtain a functional embedded systems. In particular, they shall be able to compare different models of computations and feasible techniques for system-level design. They shall be able to judge in which areas of embedded system design specific risks exist.</p>  |               |  |                     |
| <b>Personale Kompetenzen</b>                  |  |               |  |                     |
| <i>Sozialkompetenz</i>                        | Students are able to solve similar problems alone or in a group and to present the results accordingly.  |               |  |                     |
| <i>Selbstständigkeit</i>                      | Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.  |               |  |                     |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |               |  |                     |
| <b>Leistungspunkte</b>                        | 6  |               |  |                     |
| <b>Studienleistung</b>                        | <b>Verpflichtend</b>   | <b>Anteil</b> | <b>Art der Studienleistung</b>                 | <b>Beschreibung</b> |
|   | Ja   | 10 %          | Fachtheoretisch-fachpraktische Studienleistung |                     |
| <b>Prüfung</b>                                | Klausur  |               |  |                     |
| <b>Prüfungsdauer und -umfang</b>              | 90 Minuten, Inhalte der Vorlesung und Übungen  |               |  |                     |
| <b>Zuordnung zu folgenden Curricula</b>       | <p>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br/>                     Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br/>                     Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br/>                     Elektrotechnik: Kernqualifikation: Wahlpflicht<br/>                     Engineering Science: Vertiefung Mechatronik: Wahlpflicht<br/>                     Flugzeug-Systemtechnik: Vertiefung Avionik und eingebettete Systeme: Wahlpflicht<br/>                     General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     General Engineering Science (7 Semester): Vertiefung Mechatronik: Wahlpflicht<br/>                     Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br/>                     Mechatronik: Vertiefung Systementwurf: Wahlpflicht<br/>                     Mechatronik: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht<br/>                     Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht</p> |               |  |                     |

| <b>Lehrveranstaltung L0805: Embedded Systems</b> |   |
|--|---|
| <b>Typ</b>                                       | Vorlesung   |
| <b>SWS</b>                                       | 3   |
| <b>LP</b>  | 4   |
| <b>Arbeitsaufwand in Stunden</b>                 | Eigenstudium 78, Präsenzstudium 42  |
| <b>Dozenten</b>                                  | Prof. Heiko Falk  |
| <b>Sprachen</b>                                  | EN  |
| <b>Zeitraum</b>                                  | SoSe  |
| <b>Inhalt</b>                                    | <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Specifications and Modeling</li> <li>• Embedded/Cyber-Physical Systems Hardware</li> <li>• System Software</li> <li>• Evaluation and Validation</li> <li>• Mapping of Applications to Execution Platforms</li> <li>• Optimization</li> </ul> |
| <b>Literatur</b>                                 | <ul style="list-style-type: none"> <li>• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012., Springer, 2012.</li> </ul>   |

| <b>Lehrveranstaltung L0806: Embedded Systems</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>                                       | Gruppenübung                       |
| <b>SWS</b>                                       | 1                                  |
| <b>LP</b>  | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                 | Eigenstudium 46, Präsenzstudium 14 |
| <b>Dozenten</b>                                  | Prof. Heiko Falk                   |
| <b>Sprachen</b>                                  | EN                                 |
| <b>Zeitraum</b>                                  | SoSe                               |
| <b>Inhalt</b>                                    | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>                                 | Siehe korrespondierende Vorlesung  |

| <b>Modul M1300: Software Development</b>      |  |            |           |
|---|--|------------|-----------|
| <b>Lehrveranstaltungen</b>                    |  |            |           |
| <b>Titel</b>                                  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Softwareentwicklung (L1790)                   | Projekt-/problembasierte<br>Lehrveranstaltung  | 2          | 5         |
| Softwareentwicklung (L1789)                   | Vorlesung  | 1          | 1         |
| <b>Modulverantwortlicher</b>                  | Prof. Sibylle Schupp   |            |           |
| <b>Zulassungsvoraussetzungen</b>              | None   |            |           |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Introduction to Software Engineering</li> <li>• Programming Skills</li> <li>• Experience with Developing Small to Medium-Size Programs</li> </ul>   |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |            |           |
| <b>Fachkompetenz</b>                          |  |            |           |
| <i>Wissen</i>                                 | Students explain the fundamental concepts of agile methods, describe the process of test-driven development, and explain how continuous integration can be used in different scenarios. They give examples of selected pitfalls in software development, regarding scalability and other non-functional requirements. They write unit tests and build scripts and combine them in a corresponding integration environment. They explain major activities in requirements analysis, program comprehension, and agile project development. |            |           |
| <i>Fertigkeiten</i>                           | For a given task on a legacy system, students identify the corresponding parts in the system and select an appropriate method for understanding the details. They choose the proper approach of splitting a task in independent testable and extensible pieces and, thus, solve the task with proper methods for quality assurance. They design tests for legacy systems, create automated builds, and find errors at different levels. They integrate the resulting artifacts in a continuous development environment                   |            |           |
| <b>Personale Kompetenzen</b>                  |  |            |           |
| <i>Sozialkompetenz</i>                        | Students discuss different design decisions in a group. They defend their solutions orally. They communicate in English.   |            |           |
| <i>Selbstständigkeit</i>                      | Using accompanying tools, students can assess their level of knowledge continuously and adjust it appropriately. Within limits, they can set their own learning goals. Upon successful completion, students can identify and formulate concrete problems of software systems and propose solutions. Within this field, they can conduct independent studies to acquire the necessary competencies. They can devise plans to arrive at new solutions or assess existing ones.   |            |           |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 138, Präsenzstudium 42  |            |           |
| <b>Leistungspunkte</b>                        | 6  |            |           |
| <b>Studienleistung</b>                        | Keine  |            |           |
| <b>Prüfung</b>                                | Fachtheoretisch-fachpraktische Arbeit  |            |           |
| <b>Prüfungsdauer und -umfang</b>              | Software   |            |           |
| <b>Zuordnung zu folgenden Curricula</b>       | Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht   |            |           |

| <b>Lehrveranstaltung L1790: Software Development</b> |   |
|--|---|
| <b>Typ</b>   | Projekt-/problembasierte Lehrveranstaltung  |
| <b>SWS</b>   | 2   |
| <b>LP</b>  | 5   |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 122, Präsenzstudium 28   |
| <b>Dozenten</b>                                      | Prof. Sibylle Schupp  |
| <b>Sprachen</b>                                      | EN  |
| <b>Zeitraum</b>                                      | SoSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>   |
| <b>Literatur</b>                                     | <p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p> |

| <b>Lehrveranstaltung L1789: Software Development</b> |   |
|--|---|
| <b>Typ</b>   | Vorlesung   |
| <b>SWS</b>   | 1   |
| <b>LP</b>  | 1   |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 16, Präsenzstudium 14  |
| <b>Dozenten</b>                                      | Prof. Sibylle Schupp  |
| <b>Sprachen</b>                                      | EN  |
| <b>Zeitraum</b>                                      | SoSe  |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>   |
| <b>Literatur</b>                                     | <p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p> |

## Fachmodule der Vertiefung II. Mathematik und Ingenieurwissenschaften

In dieser Vertiefung geht es vor allem um Mathematik unter Verwendung von Computern. Aufbauend auf den in den ersten vier Semestern vermittelten Grundlagen der höheren Mathematik werden weitergehende Veranstaltungen aus den Bereichen Analysis, Geometrie, Kombinatorik und Numerik angeboten.

| <b>Modul M0662: Numerical Mathematics I</b>   |  |
|---|--|
| <b>Lehrveranstaltungen</b>                    |  |
| <b>Titel</b>                                  | <b>Typ</b> <b>SWS</b> <b>LP</b>  |
| Numerische Mathematik I (L0417)               | Vorlesung                      2                      3  |
| Numerische Mathematik I (L0418)               | Gruppenübung                      2                      3   |
| <b>Modulverantwortlicher</b>                  | Prof. Sabine Le Borne  |
| <b>Zulassungsvoraussetzungen</b>              | None   |
| <b>Empfohlene Vorkenntnisse</b>               | <ul style="list-style-type: none"> <li>• Mathematik I + II for Engineering Students (german or english) <b>or</b> Analysis &amp; Linear Algebra I + II for Technomathematicians</li> <li>• basic MATLAB knowledge</li> </ul>   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                          | Students are able to   |
| <i>Wissen</i>                                 | <ul style="list-style-type: none"> <li>• name numerical methods for interpolation, integration, least squares problems, eigenvalue problems, nonlinear root finding problems and to explain their core ideas,</li> <li>• repeat convergence statements for the numerical methods,</li> <li>• explain aspects for the practical execution of numerical methods with respect to computational and storage complexity.</li> </ul> |
| <i>Fertigkeiten</i>                           | Students are able to <ul style="list-style-type: none"> <li>• implement, apply and compare numerical methods using MATLAB,</li> <li>• justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm,</li> <li>• select and execute a suitable solution approach for a given problem.</li> </ul>  |
| <b>Personale Kompetenzen</b>                  | Students are able to   |
| <i>Sozialkompetenz</i>                        | <ul style="list-style-type: none"> <li>• work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.</li> </ul>  |
| <i>Selbstständigkeit</i>                      | Students are capable <ul style="list-style-type: none"> <li>• to assess whether the supporting theoretical and practical exercises are better solved individually or in a team,</li> <li>• to assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>   |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56  |
| <b>Leistungspunkte</b>                        | 6  |
| <b>Studienleistung</b>                        | Keine  |
| <b>Prüfung</b>                                | Klausur  |
| <b>Prüfungsdauer und -umfang</b>              | 90 Minuten   |
|   | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau,   |

|   |  |
|---|--|
| <b>Zuordnung zu folgenden Curricula</b> | Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht<br>Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht<br>Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Data Science: Kernqualifikation: Pflicht<br>Elektrotechnik: Kernqualifikation: Wahlpflicht<br>Engineering Science: Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Wahlpflicht<br>General Engineering Science (7 Semester): Kernqualifikation: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht<br>Maschinenbau: Vertiefung Theoretischer Maschinenbau: Wahlpflicht<br>Maschinenbau: Vertiefung Theoretischer Maschinenbau: Pflicht<br>Maschinenbau: Vertiefung Energietechnik: Wahlpflicht<br>Theoretischer Maschinenbau: Technischer Ergänzungskurs Kernfächer: Wahlpflicht<br>Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht |
|---|--|

| Lehrveranstaltung L0417: Numerical Mathematics I |   |
|--|---|
| <b>Typ</b>                                       | Vorlesung   |
| <b>SWS</b>                                       | 2   |
| <b>LP</b>  | 3   |
| <b>Arbeitsaufwand in Stunden</b>                 | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>                                  | Prof. Sabine Le Borne   |
| <b>Sprachen</b>                                  | EN  |
| <b>Zeitraum</b>                                  | WiSe  |
| <b>Inhalt</b>                                    | <ol style="list-style-type: none"> <li>1. Error analysis: Number representation, error types, conditioning and stability</li> <li>2. Interpolation: polynomial and spline interpolation</li> <li>3. Numerical integration and differentiation: order, Newton-Cotes formula, error estimates, Gaussian quadrature, adaptive quadrature, difference formulas</li> <li>4. Linear systems: LU and Cholesky factorization, matrix norms, conditioning</li> <li>5. Linear least squares problems: normal equations, Gram-Schmidt and Householder orthogonalization, singular value decomposition, regularization</li> <li>6. Eigenvalue problems: power iteration, inverse iteration, QR algorithm</li> <li>7. Nonlinear systems of equations: Fixed point iteration, root-finding algorithms for real-valued functions, Newton and Quasi-Newton methods for systems</li> </ol> |
| <b>Literatur</b>                                 | <ul style="list-style-type: none"> <li>• Stoer/Bulirsch: Numerische Mathematik 1, Springer</li> <li>• Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer</li> </ul>   |

| <b>Lehrveranstaltung L0418: Numerical Mathematics I</b> |   |
|---|---|
| <b>Typ</b>  | Gruppenübung                                |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>                        | Eigenstudium 62, Präsenzstudium 28          |
| <b>Dozenten</b>   | Prof. Sabine Le Borne, Dr. Jens-Peter Zemke |
| <b>Sprachen</b>   | EN  |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung           |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung           |

## Modul M0854: Mathematik IV

### Lehrveranstaltungen

| Titel   | Typ          | SWS | LP |
|---|--------------|-----|----|
| Differentialgleichungen 2 (Partielle Differentialgleichungen) (L1043) | Vorlesung    | 2   | 1  |
| Differentialgleichungen 2 (Partielle Differentialgleichungen) (L1044) | Gruppenübung | 1   | 1  |
| Differentialgleichungen 2 (Partielle Differentialgleichungen) (L1045) | Hörsaalübung | 1   | 1  |
| Komplexe Funktionen (L1038)   | Vorlesung    | 2   | 1  |
| Komplexe Funktionen (L1041)   | Gruppenübung | 1   | 1  |
| Komplexe Funktionen (L1042)   | Hörsaalübung | 1   | 1  |

|   |  |
|---|--|
| <b>Modulverantwortlicher</b>                  | Prof. Anusch Taraz   |
| <b>Zulassungsvoraussetzungen</b>              | Keine  |
| <b>Empfohlene Vorkenntnisse</b>               | Mathematik I - III   |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                          |  |
| <i>Wissen</i>                                 | <ul style="list-style-type: none"> <li>Studierende können die grundlegenden Begriffe der Mathematik IV benennen und anhand von Beispielen erklären.</li> <li>Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</li> <li>Sie kennen Beweisstrategien und können diese wiedergeben.</li> </ul>   |
| <i>Fertigkeiten</i>                           | <ul style="list-style-type: none"> <li>Studierende können Aufgabenstellungen aus der Mathematik IV mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.</li> <li>Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbständig zu erschließen und können diese verifizieren.</li> <li>Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.</li> </ul> |
| <b>Personale Kompetenzen</b>                  |  |
| <i>Sozialkompetenz</i>                        | <ul style="list-style-type: none"> <li>Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache.</li> <li>Sie können dabei insbesondere neue Konzepte adressatengerecht kommunizieren und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen.</li> </ul>  |
| <i>Selbstständigkeit</i>                      | <ul style="list-style-type: none"> <li>Studierende können eigenständig ihr Verständnis komplexer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen.</li> <li>Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume zielgerichtet an schwierigen Problemstellungen zu arbeiten.</li> </ul>   |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 68, Präsenzstudium 112  |
| <b>Leistungspunkte</b>                        | 6  |
| <b>Studienleistung</b>                        | Keine  |
| <b>Prüfung</b>                                | Klausur  |
| <b>Prüfungsdauer und -umfang</b>              | 60 min (Komplexe Funktionen) + 60 min (Differentialgleichungen 2)  |
|   | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Elektrotechnik: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau,   |

|   |  |
|---|--|
| <b>Zuordnung zu folgenden Curricula</b> | Schwerpunkt Mechatronik: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Schiffbau: Pflicht<br>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Wahlpflicht<br>Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Elektrotechnik: Kernqualifikation: Pflicht<br>Engineering Science: Vertiefung Elektrotechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Schiffbau: Pflicht<br>Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht<br>Maschinenbau: Vertiefung Mechatronik: Pflicht<br>Maschinenbau: Vertiefung Theoretischer Maschinenbau: Wahlpflicht<br>Maschinenbau: Vertiefung Theoretischer Maschinenbau: Pflicht<br>Mechatronik: Kernqualifikation: Pflicht<br>Schiffbau: Kernqualifikation: Pflicht<br>Theoretischer Maschinenbau: Technischer Ergänzungskurs Kernfächer: Wahlpflicht |
|---|--|

| <b>Lehrveranstaltung L1043: Differentialgleichungen 2 (Partielle Differentialgleichungen)</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 1  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 2, Präsenzstudium 28  |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH  |
| <b>Sprachen</b>   | DE   |
| <b>Zeitraum</b>   | SoSe   |
| <b>Inhalt</b>   | Grundzüge der Theorie und Numerik partieller Differentialgleichungen <ul style="list-style-type: none"> <li>• Beispiele für partielle Differentialgleichungen</li> <li>• quasilineare Differentialgleichungen erster Ordnung</li> <li>• Normalformen linearer Differentialgleichungen zweiter Ordnung</li> <li>• harmonische Funktionen und Maximumprinzip</li> <li>• Maximumprinzip für die Wärmeleitungsgleichung</li> <li>• Wellengleichung</li> <li>• Lösungsformel nach Liouville</li> <li>• spezielle Funktionen</li> <li>• Differenzenverfahren</li> <li>• finite Elemente</li> </ul> |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>  |

| <b>Lehrveranstaltung L1044: Differentialgleichungen 2 (Partielle Differentialgleichungen)</b> |   |
|---|---|
| <b>Typ</b>  | Gruppenübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | SoSe  |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung             |

| <b>Lehrveranstaltung L1045: Differentialgleichungen 2 (Partielle Differentialgleichungen)</b> |   |
|---|---|
| <b>Typ</b>  | Hörsaalübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>   | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | SoSe  |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung             |

| <b>Lehrveranstaltung L1038: Komplexe Funktionen</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 1  |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 2, Präsenzstudium 28  |
| <b>Dozenten</b>                                     | Dozenten des Fachbereiches Mathematik der UHH  |
| <b>Sprachen</b>                                     | DE   |
| <b>Zeitraum</b>                                     | SoSe   |
| <b>Inhalt</b>                                       | <p>Grundzüge der Funktionentheorie</p> <ul style="list-style-type: none"> <li>• Funktionen einer komplexen Variable</li> <li>• Komplexe Differentiation</li> <li>• Konforme Abbildungen</li> <li>• Komplexe Integration</li> <li>• Cauchyscher Hauptsatz</li> <li>• Cauchysche Integralformel</li> <li>• Taylor- und Laurent-Reihenentwicklung</li> <li>• Singularitäten und Residuen</li> <li>• Integraltransformationen: Fourier und Laplace-Transformation</li> </ul> |
| <b>Literatur</b>                                    | <ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>  |

| <b>Lehrveranstaltung L1041: Komplexe Funktionen</b> |   |
|---|---|
| <b>Typ</b>  | Gruppenübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>                                     | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>                                     | DE  |
| <b>Zeitraum</b>                                     | SoSe  |
| <b>Inhalt</b>                                       | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>                                    | Siehe korrespondierende Vorlesung             |

| <b>Lehrveranstaltung L1042: Komplexe Funktionen</b> |   |
|---|---|
| <b>Typ</b>  | Hörsaalübung                                  |
| <b>SWS</b>  | 1   |
| <b>LP</b>   | 1   |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 16, Präsenzstudium 14            |
| <b>Dozenten</b>                                     | Dozenten des Fachbereiches Mathematik der UHH |
| <b>Sprachen</b>                                     | DE  |
| <b>Zeitraum</b>                                     | SoSe  |
| <b>Inhalt</b>                                       | Siehe korrespondierende Vorlesung             |
| <b>Literatur</b>                                    | Siehe korrespondierende Vorlesung             |

| <b>Modul M0651: Rechnergestützte Geometrie</b> |  |            |           |
|--|--|------------|-----------|
| <b>Lehrveranstaltungen</b>                     |  |            |           |
| <b>Titel</b>                                   | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Rechnergestützte Geometrie (L0393)             | Vorlesung  | 2          | 4         |
| Rechnergestützte Geometrie (L0394)             | Gruppenübung   | 2          | 2         |
| <b>Modulverantwortlicher</b>                   | Dr. Prashant Batra   |            |           |
| <b>Zulassungsvoraussetzungen</b>               | Keine  |            |           |
| <b>Empfohlene Vorkenntnisse</b>                | <p>Lineare Algebra und Analytische Geometrie wie aus der schulischen Oberstufe bekannt</p> <p>(Rechnen mit Vektoren u. Determinanten, Deutung von Skalarprodukt, Kreuzprodukt, Darstellung v. Geraden/Ebenen, Satz d. Pythagoras, Cosinus-Satz, Satz d. Thales, Projektionen/Einbettungen)</p> <p>Grundlegende Datenstrukturen (Bäume, binäre Bäume, Suchbäume, balancierte binäre Bäume, Verkettete Listen)</p> <p>Definition eines Graphen</p> |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b>  | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |            |           |
| <b>Fachkompetenz</b>                           | <p>Studierende können die grundlegenden Begriffe der Rechnergestützten Geometrie benennen, mathematisch exakt beschreiben und anhand von Beispielen erklären.</p>  |            |           |
| <i>Wissen</i>                                  | Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.  |            |           |
| <i>Fertigkeiten</i>                            | Studierende können Aufgabenstellungen aus der Rechnergestützten Geometrie mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.   |            |           |
| <b>Personale Kompetenzen</b>                   | <p>Studierende sind in der Lage, mit den Teilnehmerinnen und Teilnehmer ihre eigenen algorithmischen Vorschläge zur Lösung der vorgestellten Probleme zu erörtern.</p>   |            |           |
| <i>Sozialkompetenz</i>                         | Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache.   |            |           |
| <i>Selbstständigkeit</i>                       | Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.   |            |           |
| <b>Arbeitsaufwand in Stunden</b>               | Eigenstudium 124, Präsenzstudium 56  |            |           |
| <b>Leistungspunkte</b>                         | 6  |            |           |
| <b>Studienleistung</b>                         | Keine  |            |           |
| <b>Prüfung</b>                                 | Mündliche Prüfung  |            |           |
| <b>Prüfungsdauer und -umfang</b>               | 30 min   |            |           |
| <b>Zuordnung zu folgenden Curricula</b>        | <p>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht</p> <p>Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht</p> <p>Computer Science: Vertiefung Computermathematik: Wahlpflicht</p> <p>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht</p>   |            |           |

| Lehrveranstaltung L0393: Rechnergestützte Geometrie |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 4  |
| <b>Arbeitsaufwand in Stunden</b>                    | Eigenstudium 92, Präsenzstudium 28   |
| <b>Dozenten</b>                                     | Dr. Prashant Batra   |
| <b>Sprachen</b>                                     | DE   |
| <b>Zeitraum</b>                                     | WiSe   |
| <b>Inhalt</b>                                       | <p>Konstruktion einer konvexen Hülle von n Punkten</p> <p>Triangulierung eines schlichten Polygons</p> <p>Konstruktion einer Delaunay-Triangulation, eines Voronoi-Diagramms</p> <p>Algorithmen und Datenstrukturen zum Bestimmen eines Arrangements, eines Ham-Sandwich-Cuts.</p> <p>des Schnitts von Halbebenen, der Optimierung eines linearen Funktional.</p> <p>Effiziente Bestimmung aller Schnittpunkte von (orthogonalen) Streckensegmenten</p> <p>Approximative Berechnung des Durchmessers einer Punktmenge</p> <p>Inkrementelle randomisierte Algorithmen</p> <p>Grundlagen der Gitterpunktlehre, LLL-Algorithmus und Anwendungen in der ganzzahligen Optimierung</p> <p>Grundlagen der Bewegungsplanung</p>  |
| <b>Literatur</b>                                    | <p>Computational Geometry Algorithms and Applications Authors:</p> <ul style="list-style-type: none"> <li>• Prof. Dr. Mark de Berg,</li> <li>• Dr. Otfried Cheong,</li> <li>• Dr. Marc van Kreveld,</li> <li>• Prof. Dr. Mark Overmars</li> </ul> <p>Springer e-Book: <a href="http://dx.doi.org/10.1007/978-3-540-77974-2">http://dx.doi.org/10.1007/978-3-540-77974-2</a></p> <p style="text-align: right;">Algorithmische Geometrie : Grundlagen, Methoden, Anwendungen / Rolf Klein</p> <p><b>Verfasser:</b> Klein, Rolf</p> <p><b>Ausgabe:</b> 2., vollst. überarb. Aufl.</p> <p><b>Erschienen:</b> Berlin [u.a.] : Springer, 2005</p> <p><b>Umfang:</b> XI, 392 S. : graph. Darst.</p> <p>Springer e-Book: <a href="http://dx.doi.org/10.1007/3-540-27619-X">http://dx.doi.org/10.1007/3-540-27619-X</a></p> <p>O'Rourke, Joseph</p> <p>Computational geometry in C. (English) Zbl 0816.68124</p> <p>Cambridge: Univ. Press. ix, 346 p. \$ 24.95; £16.95 /sc; \$ 59.95; £35.00 /hc (1994).</p> <p><b>ISBN:</b> 0-521-44034-3 ; 0-521-44592-2</p> <p style="text-align: right;"><b>Computational geometry</b> : an introduction / Franco P. Preparata; Michael Ian Shamos</p> <p>Preparata, Franco P. ; Shamos, Michael Ian</p> <p>Corr. and expanded 2. printing.</p> <p>New York [u.a.] : Springer, 1988</p> <p>XIV, 398 S. : graph. Darst.</p> <p>Texts and monographs in computer science</p> <p>3-540-96131-3</p> <p>0-387-96131-3</p> |

|  |   |
|--|---|
|  | Devadoss, Satyan L.; O'Rourke, Joseph<br>Discrete and computational geometry. (English) Zbl 1232.52001<br>Princeton, NJ: Princeton University Press (ISBN 978-0-691-14553-2/hbk; 978-1-400-83898-1/ebook). xi, 255 p.<br><br>ISBN: 978-3-540-77973-5 (Print) 978-3-540-77974-2 (Online) |
|--|---|

| <b>Lehrveranstaltung L0394: Rechnergestützte Geometrie</b> |                                    |
|--|------------------------------------|
| <b>Typ</b>   | Gruppenübung                       |
| <b>SWS</b>   | 2                                  |
| <b>LP</b>  | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                           | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>  | Dr. Prashant Batra                 |
| <b>Sprachen</b>  | DE                                 |
| <b>Zeitraum</b>  | WiSe                               |
| <b>Inhalt</b>  | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>   | Siehe korrespondierende Vorlesung  |

| Modul M0941: Kombinatorische Strukturen und Algorithmen |   |
|---|---|
| <b>Lehrveranstaltungen</b>                              |   |
| <b>Titel</b>  | <b>Typ</b> <b>SWS</b> <b>LP</b>   |
| Kombinatorische Strukturen und Algorithmen (L1100)      | Vorlesung                      3                      4   |
| Kombinatorische Strukturen und Algorithmen (L1101)      | Gruppenübung                      1                      2  |
| <b>Modulverantwortlicher</b>                            | Prof. Anusch Taraz  |
| <b>Zulassungsvoraussetzungen</b>                        | Keine   |
| <b>Empfohlene Vorkenntnisse</b>                         | <ul style="list-style-type: none"> <li>• Mathematik I + II</li> <li>• Diskrete Algebraische Strukturen</li> <li>• Graphentheorie und Optimierung</li> </ul>   |
| <b>Modulziele/ angestrebte Lernergebnisse</b>           | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |
| <b>Fachkompetenz</b>                                    |   |
| <i>Wissen</i>   | <ul style="list-style-type: none"> <li>• Studierende können die grundlegenden Begriffe der Kombinatorik und Algorithmik benennen und anhand von Beispielen erklären.</li> <li>• Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</li> <li>• Sie kennen Beweisstrategien und können diese wiedergeben.</li> </ul>   |
| <i>Fertigkeiten</i>                                     | <ul style="list-style-type: none"> <li>• Studierende können Aufgabenstellungen aus der Kombinatorik und Algorithmik mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.</li> <li>• Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.</li> <li>• Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.</li> </ul> |
| <b>Personale Kompetenzen</b>                            |   |
| <i>Sozialkompetenz</i>                                  | <ul style="list-style-type: none"> <li>• Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache.</li> <li>• Sie können dabei insbesondere neue Konzepte adressatengerecht kommunizieren und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen.</li> </ul>   |
| <i>Selbstständigkeit</i>                                | <ul style="list-style-type: none"> <li>• Studierende können eigenständig ihr Verständnis komplexer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen.</li> <li>• Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume zielgerichtet an schwierigen Problemstellungen zu arbeiten.</li> </ul>  |
| <b>Arbeitsaufwand in Stunden</b>                        | Eigenstudium 124, Präsenzstudium 56   |
| <b>Leistungspunkte</b>                                  | 6   |
| <b>Studienleistung</b>                                  | Keine   |
| <b>Prüfung</b>  | Mündliche Prüfung   |
| <b>Prüfungsdauer und -umfang</b>                        | 30 min  |
| <b>Zuordnung zu folgenden Curricula</b>                 | Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Data Science: Kernqualifikation: Wahlpflicht  |

|  |  |
|--|--|
|  | Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften:<br>Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Technomathematik: Vertiefung I. Mathematik: Wahlpflicht |
|--|--|

| Lehrveranstaltung L1100: Kombinatorische Strukturen und Algorithmen |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 3   |
| <b>LP</b>   | 4   |
| <b>Arbeitsaufwand in Stunden</b>                                    | Eigenstudium 78, Präsenzstudium 42  |
| <b>Dozenten</b>   | Prof. Anusch Taraz  |
| <b>Sprachen</b>   | DE/EN   |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>• Zählprobleme</li> <li>• Strukturelle Graphentheorie</li> <li>• Analyse von Algorithmen</li> <li>• Extremale Kombinatorik</li> <li>• Zufällige diskrete Strukturen</li> </ul>   |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• M. Aigner: Diskrete Mathematik, Vieweg, 6. Aufl., 2006</li> <li>• J. Matoušek &amp; J. Nešetřil: Diskrete Mathematik - Eine Entdeckungsreise, Springer, 2007</li> <li>• A. Steger: Diskrete Strukturen - Band 1: Kombinatorik, Graphentheorie, Algebra, Springer, 2. Aufl. 2007</li> <li>• A. Taraz: Diskrete Mathematik, Birkhäuser, 2012.</li> </ul> |

| Lehrveranstaltung L1101: Kombinatorische Strukturen und Algorithmen |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 1                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>                                    | Eigenstudium 46, Präsenzstudium 14 |
| <b>Dozenten</b>   | Prof. Anusch Taraz                 |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | WiSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

| <b>M o d u l M1242: Quantenmechanik für Studierende der Ingenieurwissenschaften</b> |   |            |   |
|---|---|------------|---|
| <b>Lehrveranstaltungen</b>  |   |            |   |
| <b>Titel</b>  | <b>Typ</b>  | <b>SWS</b> | <b>LP</b>   |
| Quantenmechanik für Studierende der Ingenieurwissenschaften (L1686)                 | Vorlesung   | 2          | 3   |
| Quantenmechanik für Studierende der Ingenieurwissenschaften (L1688)                 | Gruppenübung  | 2          | 3   |
| <b>Modulverantwortlicher</b>  | Prof. Wolfgang Hansen   |            |   |
| <b>Zulassungsvoraussetzungen</b>  | Keine   |            |   |
| <b>Empfohlene Vorkenntnisse</b>   | <ul style="list-style-type: none"> <li>• Kenntnisse in Physik insbesondere im Bereich Optik und Wellenlehre;</li> <li>• Kenntnisse in der Mathematik, insbesondere linearen Algebra, Vektorrechnung, komplexe Zahlen und Fourierentwicklung</li> </ul>  |            |   |
| <b>Modulziele/ angestrebte Lernergebnisse</b>                                       | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |            |   |
| <b>Fachkompetenz</b>  | Die Studierenden können grundlegende Begriffe und Prinzipien der Quantenmechanik beschreiben und erläutern. Sie kennen Gemeinsamkeiten und Unterschiede zur klassischen Physik und wissen, in welchen Situationen quantenmechanische Effekte erwartet werden können.  |            |   |
| <i>Wissen</i>   |   |            |   |
| <b>Fertigkeiten</b>   | Die Studierenden sind in der Lage, quantenmechanische Konzepte und Methoden auf einfache Probleme anzuwenden. Sie sind umgekehrt auch in der Lage, die Voraussetzungen und Prinzipien einfacher Anwendungen der Quantenmechanik in elektrooptischen Bauelementen nachzuvollziehen.  |            |   |
| <i>Fertigkeiten</i>   |   |            |   |
| <b>Personale Kompetenzen</b>  | Die Studierenden diskutieren den Vorlesungsstoff und präsentieren Lösungen einfacher quantenmechanischer Probleme in Kleingruppen während der Übungen.  |            |   |
| <i>Sozialkompetenz</i>  |   |            |   |
| <b>Selbstständigkeit</b>  | Die Studierenden sind in der Lage selbstständig einfache Lösungswege zu quantenmechanische Problemen zu erarbeiten. Sie sind so weit mit Konzepten der Quantenmechanik vertraut, dass sie sich selbstständig Literatur zu komplexeren Fragestellungen mit quantenmechanischem Hintergrund erarbeiten können.  |            |   |
| <i>Selbstständigkeit</i>  |   |            |   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 124, Präsenzstudium 56   |            |   |
| <b>Leistungspunkte</b>  | 6   |            |   |
| <b>Studienleistung</b>  | <b>Verpflichtend</b>  | <b>Ort</b> | <b>Art der Studienleistung</b> <b>Beschreibung</b>  |
|   | Nein  | Keiner     | Schriftliche Ausarbeitung<br>optionale Vorlage von selbst ausgearbeiteten Lösungen zu den Übungen |
| <b>Prüfung</b>  | Mündliche Prüfung   |            |   |
| <b>Prüfungsdauer und -umfang</b>  |   |            |   |
| <b>Zuordnung zu folgenden Curricula</b>   | Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Elektrotechnik: Kernqualifikation: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht |            |   |

| <b>Lehrveranstaltung L1686: Quantenmechanik für Studierende der Ingenieurwissenschaften</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>   | Prof. Wolfgang Hansen   |
| <b>Sprachen</b>   | DE  |
| <b>Zeitraum</b>   | WiSe  |
| <b>Inhalt</b>   | <p>Diese Veranstaltung führt in grundlegende Konzepte, Methoden und Begriffe der Quantenmechanik ein, die in den Materialwissenschaften wichtig sind. Anwendungen werden anhand konkreter Beispiele aus dem Bereich elektronischer und optischer Bauelemente diskutiert.</p> <p>Zentrale Begriffe und Themen sind:</p> <p>Schrödinger-Gleichung, Wellenfunktionen, Operatoren, Eigenzustände, Eigenwerte, Quantentöpfe, harmonischer Oszillator, Tunnelprozesse, resonante Tunneldiode, Bandstruktur, Zustandsdichte, Besetzungsverteilung, Zener-Diode, stationäre Störungsrechnung am Beispiel des Quantum Confined Stark Effekts, Fermis Goldene Regel und Übergangsmatrixelemente, Heterostrukturlaser, Quantenkaskadenlaser, Vielteilchensysteme, Moleküle und Austauschwechselwirkung, Quantenbits und Quantenkryptographie</p>   |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• David J. Griffiths: "Quantenmechanik, eine Einführung", Pearson (2012), ISBN 978-3-8632-6514-4.</li> <li>• David K. Ferry: "Quantum Mechanics", IOP Publishing (1995), ISBN 0-7503-0327-1 (hbk) bzw. 0-7503-0328-X (pbk).</li> <li>• M. Jaros: " Physics and Applications of Semiconductor Microstructures ", Clarendon Press (1989), ISBN: 0-19-851994-X bzw. 0-19-853927-4 (Pbk).</li> <li>• Randy Harris, "Modernes Physik Lehr- und Übungsbuch", 2. aktualisierte Auflage, Kapitel 3-10, Pearson (2013), ISBN 978-3-86894-115-9.</li> <li>• Michael A Nielsen and Isaac L. Chuang: "Quantum Computation and Quantum Information", 10. Auflage, Cambridge University Press (2011), ISBN: 1107002176 9781107002173.</li> <li>• Hiroyuki Sagawa and Nobuaki Yoshida: "Fundamentals of Quantum Information", World Scientific Publishing (2010), ISBN-13: 978-9814324236.</li> </ul> |

| <b>Lehrveranstaltung L1688: Quantenmechanik für Studierende der Ingenieurwissenschaften</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Wolfgang Hansen              |
| <b>Sprachen</b>   | DE                                 |
| <b>Zeitraum</b>   | WiSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

| Modul M0634: Einführung in Medizintechnische Systeme |   |                |                                |                     |
|--|---|----------------|--------------------------------|---------------------|
| <b>Lehrveranstaltungen</b>                           |   |                |                                |                     |
| <b>Titel</b>   |   | <b>Typ</b>     | <b>SWS</b>                     | <b>LP</b>           |
| Einführung in Medizintechnische Systeme (L0342)      |   | Vorlesung      | 2                              | 3                   |
| Einführung in Medizintechnische Systeme (L0343)      |   | Projektseminar | 2                              | 2                   |
| Einführung in Medizintechnische Systeme (L1876)      |   | Hörsaalübung   | 1                              | 1                   |
| <b>Modulverantwortlicher</b>                         | Prof. Alexander Schlaefer   |                |                                |                     |
| <b>Zulassungsvoraussetzungen</b>                     | Keine   |                |                                |                     |
| <b>Empfohlene Vorkenntnisse</b>                      | Grundlagen Mathematik (Algebra, Analysis)<br>Grundlagen Stochastik<br>Grundlagen Programmierung, R/Matlab   |                |                                |                     |
| <b>Modulziele/ angestrebte Lernergebnisse</b>        | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |                |                                |                     |
| <b>Fachkompetenz</b>                                 | Die Studierenden können Funktionsprinzipien ausgewählter medizintechnischer Systeme (beispielsweise bildgebende Systeme, Assistenzsysteme im OP, medizintechnische Informationssysteme) erklären. Sie können einen Überblick über regulatorische Rahmenbedingungen und Standards in der Medizintechnik geben.   |                |                                |                     |
| <i>Wissen</i>  |   |                |                                |                     |
| <i>Fertigkeiten</i>                                  | Die Studierenden sind in der Lage, die Funktion eines medizintechnischen Systems im Anwendungskontext zu bewerten.  |                |                                |                     |
| <b>Personale Kompetenzen</b>                         |   |                |                                |                     |
| <i>Sozialkompetenz</i>                               | Die Studierenden können in Gruppen ein medizintechnisches Thema als Projekt beschreiben, in Teilaufgaben untergliedern und gemeinsam bearbeiten.  |                |                                |                     |
| <i>Selbstständigkeit</i>                             | Die Studierenden können ihren Wissensstand einschätzen und ihre Arbeitsergebnisse dokumentieren. Sie können die erzielten Ergebnisse kritisch bewerten und in geeigneter Weise präsentieren.  |                |                                |                     |
| <b>Arbeitsaufwand in Stunden</b>                     | Eigenstudium 110, Präsenzstudium 70   |                |                                |                     |
| <b>Leistungspunkte</b>                               | 6   |                |                                |                     |
| <b>Studienleistung</b>                               | <b>Verpflichtend</b>  | <b>Anteil</b>  | <b>Art der Studienleistung</b> | <b>Beschreibung</b> |
|  | Ja  | 10 %           | Schriftliche Ausarbeitung      |                     |
|  | Ja  | 10 %           | Referat                        |                     |
| <b>Prüfung</b>                                       | Klausur   |                |                                |                     |
| <b>Prüfungsdauer und -umfang</b>                     | 90 Minuten  |                |                                |                     |
| <b>Zuordnung zu folgenden Curricula</b>              | Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br>Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Data Science: Kernqualifikation: Wahlpflicht<br>Elektrotechnik: Kernqualifikation: Wahlpflicht<br>Engineering Science: Vertiefung Medizingenieurwesen: Pflicht<br>General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht<br>Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Ingenieurwissenschaften: Wahlpflicht<br>Medizingenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht<br>Medizingenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht<br>Medizingenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht<br>Medizingenieurwesen: Vertiefung Management und Administration: Wahlpflicht<br>Technomathematik: Vertiefung III. Ingenieurwissenschaften: Wahlpflicht |                |                                |                     |

| <b>Lehrveranstaltung L0342: Einführung in Medizintechnische Systeme</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 3  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 62, Präsenzstudium 28   |
| <b>Dozenten</b>   | Prof. Alexander Schlaefer  |
| <b>Sprachen</b>   | DE   |
| <b>Zeitraum</b>   | SoSe   |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>- Bildgebende Systeme</li> <li>- Assistenzsysteme im OP</li> <li>- Medizintechnische Sensorsysteme</li> <li>- Medizintechnische Informationssysteme</li> <li>- Regulatorische Rahmenbedingungen</li> <li>- Standards in der Medizintechnik</li> </ul> <p>Durch problembasiertes Lernen erfolgt die Vertiefung der Methoden aus der Vorlesung. Dies erfolgt in Form von Gruppenarbeit.</p> |
| <b>Literatur</b>  | Wird in der Veranstaltung bekannt gegeben.   |

| <b>Lehrveranstaltung L0343: Einführung in Medizintechnische Systeme</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Projektseminar                     |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Alexander Schlaefer          |
| <b>Sprachen</b>   | DE                                 |
| <b>Zeitraum</b>   | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

| <b>Lehrveranstaltung L1876: Einführung in Medizintechnische Systeme</b> |  |
|---|--|
| <b>Typ</b>  | Hörsaalübung   |
| <b>SWS</b>  | 1  |
| <b>LP</b>   | 1  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 16, Präsenzstudium 14   |
| <b>Dozenten</b>   | Prof. Alexander Schlaefer  |
| <b>Sprachen</b>   | DE   |
| <b>Zeitraum</b>   | SoSe   |
| <b>Inhalt</b>   | <ul style="list-style-type: none"> <li>- Bildgebende Systeme</li> <li>- Assistenzsysteme im OP</li> <li>- Medizintechnische Sensorsysteme</li> <li>- Medizintechnische Informationssysteme</li> <li>- Regulatorische Rahmenbedingungen</li> <li>- Standards in der Medizintechnik</li> </ul> <p>Durch problembasiertes Lernen erfolgt die Vertiefung der Methoden aus der Vorlesung. Dies erfolgt in Form von Gruppenarbeit.</p> |
| <b>Literatur</b>  | Wird in der Veranstaltung bekannt gegeben.   |

| Modul M0668: Algebraische Methoden in der Regelungstechnik |  |
|--|--|
| <b>Lehrveranstaltungen</b>                                 |  |
| <b>Titel</b>   | <b>Typ</b> <b>SWS</b> <b>LP</b>  |
| Algebraische Methoden in der Regelungstechnik (L0428)      | Vorlesung 2 4  |
| Algebraische Methoden in der Regelungstechnik (L0429)      | Gruppenübung 2 2   |
| <b>Modulverantwortlicher</b>                               | Dr. Prashant Batra   |
| <b>Zulassungsvoraussetzungen</b>                           | Keine  |
| <b>Empfohlene Vorkenntnisse</b>                            | Mathe I-III (Reelle Analysis, Lineare Algebra, )<br>und entweder: Einführung in die Regelungstechnik (Beschreibung u. gewünschte Eigenschaften von Systemen, Zeitbereich/Frequenzbereich)<br>oder: Diskrete Mathematik (Gruppen, Ringe, Ideale, Körper, Euklidischer Algorithmus)  |
| <b>Modulziele/ angestrebte Lernergebnisse</b>              | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |
| <b>Fachkompetenz</b>                                       | Studierende können   |
| <i>Wissen</i>  | <ul style="list-style-type: none"> <li>• Input-Output-Systeme polynomial beschreiben,</li> <li>• Faktorisierungsansätze für Übertragungsfunktionen erklären,</li> <li>• Stabilisierungsbedingungen für Systeme in coprimer stabiler Faktorisierung benennen.</li> </ul>  |
| <i>Fertigkeiten</i>  | Studierende sind in der Lage <ul style="list-style-type: none"> <li>• eine Synthese stabiler Regelkreise durchzuführen,</li> <li>• geeignete Analyse und Synthesemethoden zur Beschreibung aller stabilen Regelkreise anzuwenden sowie</li> <li>• die Erfüllung vorgegebener Leistungsmaße sicher zu stellen.</li> </ul> |
| <b>Personale Kompetenzen</b>                               |  |
| <i>Sozialkompetenz</i>                                     | Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine zu bearbeiten und die Resultate geeignet zu präsentieren.  |
| <i>Selbstständigkeit</i>                                   | Die Studierenden können ihren Wissensstand mit Hilfe klausurnaher Aufgaben kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern.   |
| <b>Arbeitsaufwand in Stunden</b>                           | Eigenstudium 124, Präsenzstudium 56  |
| <b>Leistungspunkte</b>                                     | 6  |
| <b>Studienleistung</b>                                     | Keine  |
| <b>Prüfung</b>   | Mündliche Prüfung  |
| <b>Prüfungsdauer und -umfang</b>                           | 30 min   |
| <b>Zuordnung zu folgenden Curricula</b>                    | Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Ingenieurwissenschaften: Wahlpflicht<br>Technomathematik: Vertiefung II. Informatik: Wahlpflicht                           |

| <b>Lehrveranstaltung L0428: Algebraische Methoden in der Regelungstechnik</b> |  |
|---|--|
| <b>Typ</b>  | Vorlesung  |
| <b>SWS</b>  | 2  |
| <b>LP</b>   | 4  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 92, Präsenzstudium 28   |
| <b>Dozenten</b>   | Dr. Prashant Batra   |
| <b>Sprachen</b>   | DE/EN  |
| <b>Zeitraum</b>   | SoSe   |
| <b>Inhalt</b>   | <p>- Algebraische Methoden der Regelungstechnik, polynomialer Ansatz, Faktorisierungsbeschreibung</p> <p>- Beschreibung 1-dimensionaler Regelsysteme, Synthese von (minimalen) Regelsystemen durch algebraische Interpolationsmethoden,</p> <p>- Simultane Stabilisierbarkeit</p> <p>- Parametrisierung sämtlicher stabilisierenden Regler</p> <p>- Reglerentwurf bei Polvorgabe</p> <p>- Berücksichtigung von Systemeigenschaften: Störanfälligkeit, Sensitivität.</p> <p>- Polynomiale Matrizen, Beschreibung durch Links-Faktorisierungen.</p> <p>- Euklidischer Algorithmus u. Diophantische Gleichungen über Ringen</p> <p>- Smith-McMillan Normal Form</p> <p>- Synthese von Mehrgrößensystemen durch polynomiale Methoden</p> |
| <b>Literatur</b>  | <ul style="list-style-type: none"> <li>• Vidyasagar, M.: Control system synthesis: a factorization approach. The MIT Press, Cambridge/Mass. - London, 1985.</li> <li>• Vardulakis, A.I.G.: Linear multivariable control. Algebraic analysis and synthesis methods, John Wiley &amp; Sons, Chichester, UK, 1991.</li> <li>• Chen, Chi-Tsong: Analog and digital control system design. Transfer-function, state-space, and algebraic methods. Oxford Univ. Press, 1995.</li> <li>• Kučera, V.: Analysis and Design of Discrete Linear Control Systems. Praha: Academia, 1991.</li> </ul>  |

| <b>Lehrveranstaltung L0429: Algebraische Methoden in der Regelungstechnik</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 2                                  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 32, Präsenzstudium 28 |
| <b>Dozenten</b>   | Dr. Prashant Batra                 |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

| Modul M1269: Labor Cyber-Physical Systems     |   |     |    |
|---|---|-----|----|
| Lehrveranstaltungen                           |   |     |    |
| Titel   | Typ   | SWS | LP |
| Labor Cyber-Physical Systems (L1740)          | Projekt-/problembasierte Lehrveranstaltung  | 4   | 6  |
| <b>Modulverantwortlicher</b>                  | Prof. Heiko Falk  |     |    |
| <b>Zulassungsvoraussetzungen</b>              | Keine   |     |    |
| <b>Empfohlene Vorkenntnisse</b>               | Modul "Eingebettete Systeme"  |     |    |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht   |     |    |
| <b>Fachkompetenz</b>                          | <p>Cyber-Physical Systems (CPS) stehen über Sensoren, A/D- und D/A-Wandler und Aktoren in enger Verbindung mit ihrer Umgebung. Wegen der besonderen Einsatzgebiete kommen hier hochgradig spezialisierte Sensoren, Prozessoren und Aktoren zum Einsatz, die applikationsspezifisch auf ihr jeweiliges Einsatzgebiet ausgerichtet sind. Dementsprechend existiert - im Gegensatz zum klassischen Software Engineering - eine Vielzahl unterschiedlicher Techniken zur Spezifikation von CPS.</p> <p><i>Wissen</i> In Form von rechnergestützten Versuchen mit Roboterbausätzen werden in dieser Veranstaltung die Grundzüge der Spezifikation und Modellierung von CPS vermittelt. Das Labor behandelt die Einführung in diese Systeme (Begriffsbildung, charakteristische Eigenschaften) und deren Spezifikationsprachen (models of computation, hierarchische Zustandsautomaten, Datenfluss-Modelle, Petri-Netze, imperative Techniken). Da CPS häufig Steuerungs- und Regelungsaufgaben erfüllen, wird das Labor praxisnah einfache Anwendungen aus der Regelungstechnik vermitteln. Die Versuche nutzen gängige Spezifikationswerkzeuge (MATLAB/Simulink, LabVIEW, NXC), um hiermit Cyber-Physical Systems zu modellieren, die über Sensoren und Aktoren mit ihrer Umwelt interagieren.</p> <p><i>Fertigkeiten</i> Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage, einfache CPS zu entwickeln. Sie können Wechselwirkungen zwischen einem CPS und dessen umgebenden Prozessen beurteilen, der sich aus dem Kreislauf zwischen physikalischer Umwelt, Sensor, A/D-Wandler, digitalem Prozessor, D/A-Wandler und Aktor ergibt. Die Veranstaltung versetzt die Studierenden in die Lage, Modellierungstechniken miteinander vergleichen, deren Vor- und Nachteile abwägen, und geeignete Techniken zur Systementwicklung einsetzen zu können. Sie erwerben die Fähigkeit, diese Techniken im Rahmen konkreter praktischer Aufgabenstellungen anzuwenden. Sie haben erste Erfahrungen im hardwarenahen Software-Entwurf, im Umgang mit industrierelevanten Spezifikationswerkzeugen und im Entwurf einfacher Regelungssysteme erworben.</p> |     |    |
| <b>Personale Kompetenzen</b>                  | <p><i>Sozialkompetenz</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, ähnliche Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.</p> <p><i>Selbstständigkeit</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.</p>   |     |    |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 124, Präsenzstudium 56   |     |    |
| <b>Leistungspunkte</b>                        | 6   |     |    |
| <b>Studienleistung</b>                        | Keine   |     |    |
| <b>Prüfung</b>                                | Schriftliche Ausarbeitung   |     |    |
| <b>Prüfungsdauer und -umfang</b>              | Durchführung und Beschreibung sämtlicher Versuche   |     |    |
| <b>Zuordnung zu folgenden Curricula</b>       | <p>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht<br/>                     Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br/>                     General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht<br/>                     Informatik-Ingenieurwesen: Vertiefung II. Mathematik &amp; Ingenieurwissenschaften: Wahlpflicht<br/>                     Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht</p>  |     |    |

|  |
|--|
| Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht<br>Mechatronics: Vertiefung Systementwurf: Wahlpflicht<br>Mechatronics: Technischer Ergänzungskurs: Wahlpflicht |
|--|

| <b>Lehrveranstaltung L1740: Labor Cyber-Physical Systems</b> |  |
|--|--|
| <b>Typ</b>   | Projekt-/problembasierte Lehrveranstaltung   |
| <b>SWS</b>   | 4  |
| <b>LP</b>  | 6  |
| <b>Arbeitsaufwand in Stunden</b>                             | Eigenstudium 124, Präsenzstudium 56  |
| <b>Dozenten</b>  | Prof. Heiko Falk   |
| <b>Sprachen</b>  | DE/EN  |
| <b>Zeitraum</b>  | SoSe   |
| <b>Inhalt</b>  | <ul style="list-style-type: none"> <li>• Versuch 1: Programmieren in NXC</li> <li>• Versuch 2: Programmierung des Roboters mit Matlab/Simulink</li> <li>• Programmierung des Roboters in LabVIEW</li> </ul>                    |
| <b>Literatur</b>   | <ul style="list-style-type: none"> <li>• Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012.</li> <li>• Begleitende Foliensätze</li> </ul> |

| Modul M0715: Löser für schwachbesetzte lineare Gleichungssysteme |  |              |            |           |
|--|--|--------------|------------|-----------|
| <b>Lehrveranstaltungen</b>                                       |  |              |            |           |
| <b>Titel</b>   |  | <b>Typ</b>   | <b>SWS</b> | <b>LP</b> |
| Löser für schwachbesetzte lineare Gleichungssysteme (L0583)      |  | Vorlesung    | 2          | 3         |
| Löser für schwachbesetzte lineare Gleichungssysteme (L0584)      |  | Gruppenübung | 2          | 3         |
| <b>Modulverantwortlicher</b>                                     | Prof. Sabine Le Borne  |              |            |           |
| <b>Zulassungsvoraussetzungen</b>                                 | Keine  |              |            |           |
| <b>Empfohlene Vorkenntnisse</b>                                  | <ul style="list-style-type: none"> <li>• Mathematik I + II für Ingenieurstudierende (deutsch oder englisch) oder Analysis &amp; Lineare Algebra I + II für Technomathematiker</li> <li>• Programmierkenntnisse in C</li> </ul>   |              |            |           |
| <b>Modulziele/ angestrebte Lernergebnisse</b>                    | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |              |            |           |
| <b>Fachkompetenz</b>   | Studierende können   |              |            |           |
| <i>Wissen</i>  | <ul style="list-style-type: none"> <li>• klassische und moderne Iterationsverfahren und deren Zusammenhänge untereinander benennen,</li> <li>• Konvergenzaussagen zu Iterationsverfahren wiedergeben,</li> <li>• Aspekte der effizienten Implementierung von Iterationsverfahren erklären.</li> </ul>  |              |            |           |
| <i>Fertigkeiten</i>  | Studierende sind in der Lage, <ul style="list-style-type: none"> <li>• Iterationsverfahren zu implementieren, anzuwenden und zu vergleichen,</li> <li>• das Konvergenzverhalten von Iterationsverfahren zu analysieren und gegebenenfalls Konvergenzraten zu berechnen.</li> </ul>   |              |            |           |
| <b>Personale Kompetenzen</b>                                     | Studierende können   |              |            |           |
| <i>Sozialkompetenz</i>   | <ul style="list-style-type: none"> <li>• in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen.</li> </ul>   |              |            |           |
| <i>Selbstständigkeit</i>   | Studierende sind fähig, <ul style="list-style-type: none"> <li>• selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen,</li> <li>• mit ausreichender Ausdauer komplexe Problemstellungen über längere Zeiträume zu bearbeiten,</li> <li>• ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.</li> </ul> |              |            |           |
| <b>Arbeitsaufwand in Stunden</b>                                 | Eigenstudium 124, Präsenzstudium 56  |              |            |           |
| <b>Leistungspunkte</b>   | 6  |              |            |           |
| <b>Studienleistung</b>   | Keine  |              |            |           |
| <b>Prüfung</b>   | Mündliche Prüfung  |              |            |           |
| <b>Prüfungsdauer und -umfang</b>                                 | 20 min   |              |            |           |
| <b>Zuordnung zu folgenden Curricula</b>                          | Computer Science: Vertiefung Computermathematik: Wahlpflicht<br>Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht<br>Data Science: Kernqualifikation: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht<br>Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht<br>Technomathematik: Vertiefung I. Mathematik: Wahlpflicht    |              |            |           |

| <b>Lehrveranstaltung L0583: Löser für schwachbesetzte lineare Gleichungssysteme</b> |   |
|---|---|
| <b>Typ</b>  | Vorlesung   |
| <b>SWS</b>  | 2   |
| <b>LP</b>   | 3   |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 62, Präsenzstudium 28  |
| <b>Dozenten</b>   | Prof. Sabine Le Borne   |
| <b>Sprachen</b>   | DE/EN   |
| <b>Zeitraum</b>   | SoSe  |
| <b>Inhalt</b>   | <ol style="list-style-type: none"> <li>1. Schwachbesetzte Matrizen: Anordnungen und Speicherformate, direkte Löser</li> <li>2. Klassische Iterationsverfahren: Grundbegriffe, Konvergenz</li> <li>3. Projektionsverfahren</li> <li>4. Krylovraumverfahren</li> <li>5. Präkonditionierung (z.B ILU)</li> <li>6. Mehrgitterverfahren</li> </ol> |
| <b>Literatur</b>  | <ol style="list-style-type: none"> <li>1. Y. Saad, Iterative methods for sparse linear systems</li> </ol>   |

| <b>Lehrveranstaltung L0584: Löser für schwachbesetzte lineare Gleichungssysteme</b> |                                    |
|---|------------------------------------|
| <b>Typ</b>  | Gruppenübung                       |
| <b>SWS</b>  | 2                                  |
| <b>LP</b>   | 3                                  |
| <b>Arbeitsaufwand in Stunden</b>  | Eigenstudium 62, Präsenzstudium 28 |
| <b>Dozenten</b>   | Prof. Sabine Le Borne              |
| <b>Sprachen</b>   | DE/EN                              |
| <b>Zeitraum</b>   | SoSe                               |
| <b>Inhalt</b>   | Siehe korrespondierende Vorlesung  |
| <b>Literatur</b>  | Siehe korrespondierende Vorlesung  |

## Fachmodule der Vertiefung III. Fachspezifische Fokussierung

Die beiden Technischen Ergänzungsfächer dienen der weiteren Vertiefung der im Wahlpflichtbereich gewählten Module.

| Modul M1562: Technischer Ergänzungskurs I für CSBS                                 |   |     |    |
|--|---|-----|----|
| Lehrveranstaltungen  |   |     |    |
| Titel  | Typ   | SWS | LP |
| <b>Modulverantwortlicher</b>   | Prof. Karl-Heinz Zimmermann   |     |    |
| <b>Zulassungsvoraussetzungen</b>   | Keine   |     |    |
| <b>Empfohlene Vorkenntnisse</b>  |   |     |    |
| <b>Modulziele/ angestrebte Lernergebnisse</b>                                      | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht |     |    |
| <b>Fachkompetenz</b><br><i>Wissen</i><br><i>Fertigkeiten</i>                       |   |     |    |
| <b>Personale Kompetenzen</b><br><i>Sozialkompetenz</i><br><i>Selbstständigkeit</i> |   |     |    |
| <b>Arbeitsaufwand in Stunden</b>   | Abhängig von der Wahl der Lehrveranstaltungen   |     |    |
| <b>Leistungspunkte</b>   | 6   |     |    |
| <b>Zuordnung zu folgenden Curricula</b>  | Computer Science: Vertiefung III. Fachspezifische Fokussierung: Wahlpflicht               |     |    |

| Modul M1568: Technischer Ergänzungskurs II für CSBS                                |   |     |    |
|--|---|-----|----|
| Lehrveranstaltungen  |   |     |    |
| Titel  | Typ   | SWS | LP |
| <b>Modulverantwortlicher</b>   | Prof. Karl-Heinz Zimmermann   |     |    |
| <b>Zulassungsvoraussetzungen</b>   | Keine   |     |    |
| <b>Empfohlene Vorkenntnisse</b>  |   |     |    |
| <b>Modulziele/ angestrebte Lernergebnisse</b>                                      | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht |     |    |
| <b>Fachkompetenz</b><br><i>Wissen</i><br><i>Fertigkeiten</i>                       |   |     |    |
| <b>Personale Kompetenzen</b><br><i>Sozialkompetenz</i><br><i>Selbstständigkeit</i> |   |     |    |
| <b>Arbeitsaufwand in Stunden</b>   | Abhängig von der Wahl der Lehrveranstaltungen   |     |    |
| <b>Leistungspunkte</b>   | 6   |     |    |
| <b>Zuordnung zu folgenden Curricula</b>  | Computer Science: Vertiefung III. Fachspezifische Fokussierung: Wahlpflicht               |     |    |

## Thesis

### Modul M-001: Bachelorarbeit

#### Lehrveranstaltungen

| Titel   | Typ  | SWS | LP |
|---|--|-----|----|
| <b>Modulverantwortlicher</b>                  | Professoren der TUHH   |     |    |
| <b>Zulassungsvoraussetzungen</b>              | <ul style="list-style-type: none"> <li>Laut ASPO § 21 (1):<br/>Es müssen mindestens 126 Leistungspunkte im Studiengang erworben worden sein. Über Ausnahmen entscheidet der Prüfungsausschuss.</li> </ul>  |     |    |
| <b>Empfohlene Vorkenntnisse</b>               |  |     |    |
| <b>Modulziele/ angestrebte Lernergebnisse</b> | Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht  |     |    |
| <b>Fachkompetenz</b>                          | <ul style="list-style-type: none"> <li>Studierende können die wichtigsten wissenschaftlichen Grundlagen ihres Studienfaches (Fakten, Theorien und Methoden) problembezogen auswählen, darstellen und nötigenfalls kritisch diskutieren.</li> <li>Die Studierenden können ausgehend von ihrem fachlichen Grundlagenwissen anlassbezogen auch weiterführendes fachliches Wissen erschließen und verknüpfen.</li> <li>Die Studierenden können zu einem ausgewählten Thema ihres Faches einen Forschungsstand darstellen.</li> </ul> |     |    |
| <i>Wissen</i>                                 |  |     |    |
| <i>Fertigkeiten</i>                           | <ul style="list-style-type: none"> <li>Die Studierenden können das im Studium vermittelte Grundwissen ihres Studienfaches zielgerichtet zur Lösung fachlicher Probleme einsetzen.</li> <li>Die Studierenden können mit Hilfe der im Studium erlernten Methoden Fragestellungen analysieren, fachliche Sachverhalte entscheiden und Lösungen entwickeln.</li> <li>Die Studierenden können zu den Ergebnissen ihrer eigenen Forschungsarbeit kritisch aus einer Fachperspektive Stellung beziehen.</li> </ul>                      |     |    |
| <b>Personale Kompetenzen</b>                  | <ul style="list-style-type: none"> <li>Studierende können eine wissenschaftliche Fragestellung für ein Fachpublikum sowohl schriftlich als auch mündlich strukturiert, verständlich und sachlich richtig darstellen.</li> <li>Studierende können in einer Fachdiskussion auf Fragen eingehen und sie in adressatengerechter Weise beantworten. Sie können dabei eigene Einschätzungen und Standpunkte überzeugend vertreten.</li> </ul>  |     |    |
| <i>Sozialkompetenz</i>                        |  |     |    |
| <i>Selbstständigkeit</i>                      | <ul style="list-style-type: none"> <li>Studierende können einen umfangreichen Arbeitsprozess zeitlich strukturieren und eine Fragestellung in vorgegebener Frist bearbeiten.</li> <li>Studierende können notwendiges Wissen und Material zur Bearbeitung eines wissenschaftlichen Problems identifizieren, erschließen und verknüpfen.</li> <li>Studierende können die wesentlichen Techniken des wissenschaftlichen Arbeitens in einer eigenen Forschungsarbeit anwenden.</li> </ul>  |     |    |
| <b>Arbeitsaufwand in Stunden</b>              | Eigenstudium 360, Präsenzstudium 0   |     |    |
| <b>Leistungspunkte</b>                        | 12   |     |    |
| <b>Studienleistung</b>                        | Keine  |     |    |
| <b>Prüfung</b>                                | Abschlussarbeit  |     |    |
| <b>Prüfungsdauer und -umfang</b>              | laut ASPO  |     |    |
|   | Allgemeine Ingenieurwissenschaften (7 Semester): Abschlussarbeit: Pflicht<br>Bau- und Umweltingenieurwesen: Abschlussarbeit: Pflicht<br>Bioverfahrenstechnik: Abschlussarbeit: Pflicht<br>Computer Science: Abschlussarbeit: Pflicht<br>Data Science: Abschlussarbeit: Pflicht   |     |    |

|   |  |
|---|--|
| <b>Zuordnung zu folgenden Curricula</b> | Digitaler Maschinenbau: Abschlussarbeit: Pflicht<br>Elektrotechnik: Abschlussarbeit: Pflicht<br>Energie- und Umwelttechnik: Abschlussarbeit: Pflicht<br>Engineering Science: Abschlussarbeit: Pflicht<br>General Engineering Science (7 Semester): Abschlussarbeit: Pflicht<br>Informatik-Ingenieurwesen: Abschlussarbeit: Pflicht<br>Logistik und Mobilität: Abschlussarbeit: Pflicht<br>Maschinenbau: Abschlussarbeit: Pflicht<br>Mechatronik: Abschlussarbeit: Pflicht<br>Schiffbau: Abschlussarbeit: Pflicht<br>Technomathematik: Abschlussarbeit: Pflicht<br>Teilstudiengang Lehramt Elektrotechnik-Informationstechnik: Abschlussarbeit: Pflicht<br>Teilstudiengang Lehramt Metalltechnik: Abschlussarbeit: Pflicht<br>Verfahrenstechnik: Abschlussarbeit: Pflicht |
|---|--|