

**Modulhandbuch
B.Sc. Informatik**

Ziele

Der Studiengang richtet sich an Absolventen von Gymnasien oder Fachoberschulen und setzt keine Vorkenntnisse im Bereich der Informatik voraus. Die Absolventen des Studiengangs sollen über solide Kenntnisse und Fertigkeiten im Gesamtgebiet der Informatik einschließlich ihrer Grundlagen sowie über vertiefte Kenntnisse und Kompetenzen in ausgewählten Teil- und Anwendungsgebieten verfügen.

Der Kasseler Informatik-Studiengang ist technisch orientiert, d.h. er betont Berufsfelder im Grenzgebiet zwischen Elektrotechnik und Informatik, gewährleistet aber auch für weitere Einsatzgebiete der Informatik eine adäquate Ausbildung. Die Schwerpunktsetzung erfolgt durch die Studierenden im Studienverlauf.

Der Studiengang bildet Informatiker für die Wirtschaft der Region und darüber hinaus aus. Entsprechend werden die Studierenden nicht nur fachlich, sondern auch methodisch zu einer Berufstätigkeit in der Informatik (zunächst ohne deutlichen Forschungsbezug) befähigt werden. Der Studiengang bereitet weiterhin auf den konsekutiven Masterstudiengang Informatik sowie ggf. weiterführend auf Tätigkeiten als künftige Doktoranden und Wissenschaftler vor. Die Studierenden erwerben dafür Methoden des wissenschaftlichen Arbeitens, so dass sie prinzipiell ein Masterstudium aufnehmen können.

Angestrebte Lernergebnisse

Die Absolventen sollen einerseits über solide Kenntnisse in allen zentralen und ausgewählten weiteren Teilgebieten der Informatik sowie einem Anwendungsgebiet verfügen und andererseits in der Lage sein, anspruchsvolle Probleme zu analysieren und mit Hilfe von Informatik-Methoden konstruktiv und kreativ zu lösen. Dazu müssen sie informatische Denkweisen wie prozedurales und strukturelles Denken beherrschen. Für einen flexiblen Einsatz in unterschiedlichen Berufsfeldern sollen die Absolventen in der Lage sein, sich in angemessener Zeit in neue Teilgebiete der Informatik und angrenzender Gebiete einzuarbeiten. Die Absolventen haben eine Gesamtsicht auf das Fach und können so Zusammenhänge erkennen. Im Einzelnen sollen folgende Lernergebnisse erreicht werden:

- Kenntnisse in der Informatik, in mathematischen und elektrotechnischen Grundlagen sowie in einem Anwendungsgebiet: Die Informatik-Kenntnisse umfassen ein solides Grundwissen in allen zentralen Teilgebieten sowie vertiefte Kenntnisse in ausgewählten Teilgebieten mit selbst gewählter Schwerpunktsetzung. Die Grundlagenkenntnisse schaffen die Basis für das Verständnis der Informatik-Inhalte und -methoden in Studium und späterer Weiterbildung. Mit der Schwerpunktsetzung und Wahl des Anwendungsgebietes erwerben die Studierenden in

Teilbereichen einen Kenntnisstand, der die eigene Tätigkeit (Anwendung von Informatik-Methoden, Berufstätigkeit) in diesen Gebieten ermöglicht.

- Fertigkeiten in der Anwendung eines breiten Spektrums von Informatik-Methoden: Dazu gehören unter anderem Fertigkeiten in: Programmierung, Algorithmenentwurf, Analyse und Modellbildung, Design von Hard- und/oder Softwaresystemen unter Berücksichtigung von Qualitätsanforderungen, Techniken des Software-Engineerings, Einarbeitung in existierende Hard- und/oder Softwaresysteme, Erstellen und Umsetzen formaler Spezifikationen, Korrektheits- und Effizienznachweise. Je nach Schwerpunktsetzung verfügen die Absolventen über einige dieser speziellen Fertigkeiten in vertiefter Form bzw. über weitere Fertigkeiten.
- Methodenkompetenzen: Dazu gehört die Fähigkeit zum wissenschaftlichen Arbeiten, zur selbstständigen Einarbeitung in neue Themen einschließlich der Erarbeitung englischsprachiger Literatur, zur Teamarbeit einschließlich Kommunikations-, Organisations- und Konfliktmanagementkompetenzen sowie zur mündlichen und schriftlichen Präsentation. Weiterhin besitzen die Absolventen die Fähigkeit zur kritischen Reflexion der eigenen Tätigkeit und die Bereitschaft zur Übernahme von Verantwortung für das Ergebnis der eigenen Arbeit.
- Fachübergreifende und Realisierungskompetenzen: Basis für diese Kompetenzen sind Kenntnisse zu exemplarischen Anwendungsgebieten und Beispielen für den praktischen Einsatz der Informatik-Methoden sowie Grundkenntnisse zu Wirtschaft, Recht und Managementtechniken. Insbesondere erwerben alle Studierenden Kompetenzen im Projektmanagement. Darüber hinaus arbeiten sich die Studierenden in Denk- und Arbeitsweisen der Wirtschaft, des Rechts und des Managements ein, um ein grundlegendes Verständnis für die entsprechenden Blickwinkel in ihre Arbeit einfließen lassen zu können und um sich eine breite Basis für lebenslanges Lernen zu schaffen. Die Kenntnisse werden im Studium durch eigene praktische Arbeit zu Realisierungskompetenzen ausgebaut.

Pflichtmodule

Die einzelnen Grundbereiche setzen sich aus den folgenden Modulen zusammen:

| Grundbereich | Dazugehörige Module |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Analysis für Informatiker | Analysis für Informatiker (S. 12) |
| Differenzierungsmodul | Differenzierungsmodul (S. 15) |
| Digitale Rechnerarchitekturen | Digitale Logik (S. 19) Rechnerarchitektur (S. 29) |
| Diskrete Strukturen | Diskrete Strukturen I (S. 20) Diskrete Strukturen II (S. 21) |
| Elektrotechnik/Elektronik | Elektrotechnik für Informatiker (S. 25) Grundwissen der Elektronik (S. 26) |
| Lineare Algebra | Lineare Algebra (S. 27) |
| Praktische Informatik | Betriebssysteme (S. 13) Datenbanken (S. 14) Einführung in die Künstliche Intelligenz (S. 23) |
| Programmierung | Algorithmen und Datenstrukturen (S. 11) Einführung in C (S. 22) Einführung in die Programmierung für Informatik (S. 24) |
| Software Entwicklung | Programmiermethodik (S. 28) Softwaretechnik I (S. 31) |
| Technische Informatik | Rechnernetze (S. 30) Systemprogrammierung (S. 32) |
| Theoretische Informatik | Theoretische Informatik – Berechenbarkeit und Formale Sprachen (S. 33) Theoretische Informatik – Logik (S. 34) |

Spezielle Module

| | |
|----------------------|--------|
| Berufspraxis | S. 134 |
| Projekt | S. 135 |
| Schlüsselkompetenzen | S. 136 |
| Seminar | S. 137 |

Wahlpflichtmodule

| Wahlpflicht Praktische Informatik (Fohry) | Seite |
|----------------------------------------------------------------------------------------------------------------------------|-------|
| Advanced Game Programming | 36 |
| Applikationsentwicklung für Tablet-Computer | 38 |
| Autonome Mobile Roboter | 44 |
| C++ für Fortgeschrittene | 47 |
| Compilerbau und Reverse Engineering | 48 |
| Data Mining für Technische Anwendungen | 51 |
| Design Patterns | 53 |
| Einführung in die Programmierung mit C++ | 59 |
| Einführung in Matlab/Simulink | 62 |
| Funktionale Programmierung | 67 |
| Graphische Simulation | 69 |
| Grundlagen der angewandten Kryptologie | 71 |
| Internet-Suchmaschinen | 84 |
| Knowledge Discovery | 86 |
| Parallelverarbeitung I | 103 |
| Parallelverarbeitung II | 104 |
| Praktikum 3D Game Programmierung | 105 |
| Praktikum Internet-Suchmaschinen | 108 |
| Praktikum Knowledge Discovery | 109 |
| Praktikum kooperative verteilte Robotersysteme | 110 |
| Praktikum VR Game Programmierung | 113 |
| Seminar in den Fachgebieten Programmiersprachen/-methodik, Software Engineering, Verteilte Systeme, Wissensverarbeitung | 137 |
| Sicherheit in Kommunikationsnetzen | 118 |
| Soft Computing | 121 |
| Softwareergonomie | 122 |
| Softwarequalität | 124 |
| Techniken und Dienste des Internets | 126 |
| Verteilte Systeme - Architekturen und Dienste | 130 |

| Wahlpflicht Technische Informatik (Böröcsök) | Seite |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 3D Modellierung | 35 |
| Ausgewählte Kapitel der Kommunikationstechnik | 41 |
| Ausgewählte Kapitel der Rechnerarchitektur | 42 |
| Ausgewählte Kapitel der Rechnerarchitektur und Mikroprozessortechnik | 43 |
| Avatare in Serious Games | 45 |
| Computergraphik | 50 |
| Digitale Signalverarbeitung mit integrierten Schaltungen | 54 |
| Digitale Systeme | 55 |
| Echtzeitsysteme | 56 |
| Immersive 3D Welten | 76 |
| Industrielle Netzwerke | 77 |
| Intelligent Humanoid Robots I | 82 |
| Intelligente Technische Systeme | 83 |
| Java Code-Camp Context Awareness I | 85 |
| Laborpraktikum Mobile Telefonsysteme | 88 |
| Laborpraktikum Rechnernetze | 89 |
| Mathematische Modelle zur MTTF und PFD Analyse | 94 |
| Mikroprozessortechnik – Labor | 97 |
| Mikroprozessortechnik und eingebettete Systeme 1 | 98 |
| Mikroprozessortechnik und eingebettete Systeme 2 | 99 |
| Mobile Computing | 100 |
| Praktikum Digitaltechnik | 106 |
| Praktikum Intelligente Eingebettete Systeme | 107 |
| Programmierung mobiler Geräte (Android und iOS) | 114 |
| Prozessleittechnik | 115 |
| Risikobewertung von Rechnerarchitekturen I | 117 |
| Seminar in den Fachgebieten Digitaltechnik, Intelligente Eingebettete Systeme, Kommunikationstechnik, Rechnerarchitektur und Systemprogrammierung, Technische Informatik | 137 |
| Signalverarbeitung mit Mikroprozessoren I | 120 |
| SPS Programmierung nach IEC 61131-3 | 125 |
| VHDL – Kurs | 132 |
| VHDL – Praktikum | 133 |

| Wahlpflicht Schwerpunkt (Fohry) | Seite |
|----------------------------------------------------------------------------------------|--------------|
| Module aus Wahlpflicht Praktische Informatik, Wahlpflicht Technische Informatik, oder: | |
| Einführung in die formale Verifikation | 58 |
| Entwurf und Analyse von Algorithmen | 64 |
| Formale Sprachen und Automaten I | 66 |
| Grundlagen der Elektrotechnik II für Informatiker | 72 |
| Grundlagen der Stochastik | 75 |
| Komplexitätstheorie | 87 |
| Nutzungsorientierte Gestaltung | 102 |
| Reduktionssysteme I | 116 |

Anwendungsgebiet Computational Mathematics (Koepf) Seite

Basis Anwendungsgebiet

Grundlagen der Algebra und Computeralgebra 70

Anwendungsgebiet (12 CP nach Wahl)

Algebra 37

Vertiefungsvorlesungen Mathematik (Mathematik: B9) 131

Anwendungsgebiet Computergraphik (Wloka) Seite

Für Basis Anwendungsgebiet 6 CP, für Anwendungsgebiet 12 CP wählen:

| | |
|---------------------------------------|-----|
| 3D Modellierung | 35 |
| Advanced Game Programming | 36 |
| Avatare in Serious Games | 45 |
| C++ für Fortgeschrittene | 47 |
| Computergraphik | 50 |
| Graphische Simulation | 69 |
| Immersive 3D Welten | 76 |
| Mensch-Maschine-Systeme 1 | 95 |
| Mensch-Maschine-Systeme 2 | 96 |
| Praktikum 3D Game Programmierung | 105 |
| Praktikum Mensch-Maschine-Interaktion | 112 |
| Praktikum VR Game Programmierung | 113 |

Anwendungsgebiet Embedded Intelligence (Sick) Seite

Für Basis Anwendungsgebiet 6 CP, für Anwendungsgebiet 12 CP wählen:

| | |
|-----------------------------------------------------------|-----|
| Intelligente Technische Systeme | 83 |
| Autonome Mobile Roboter | 44 |
| Computational Intelligence in der Automatisierung oder | 49 |
| Soft Computing | 121 |
| Data Mining für Technische Anwendungen oder | 51 |
| Knowledge Discovery | 86 |
| Digitale Systeme oder | 55 |
| Signalverarbeitung mit Mikroprozessoren I | 120 |
| Echtzeitsysteme | 56 |
| Grundlagen der Regelungstechnik | 74 |
| Intelligent Humanoid Robots I | 82 |
| Java Code-Camp Context Awareness I | 85 |
| Praktikum Intelligente Eingebettete Systeme | 107 |
| Praktikum kooperative verteilte Robotersysteme | 110 |

Der Besuch der Vorlesung „*Intelligente Technische Systeme*“ als Basis wird empfohlen, er ist aber nicht verpflichtend.

Von den Studierenden müssen folgende Restriktionen beachtet werden:

- Maximal 6 CP können über Praktika abgedeckt werden.
- Es kann im Anwendungsgebiet nur entweder die Vorlesung „*Computational Intelligence in der Automatisierung*“ oder die Vorlesung „*Soft Computing*“ angerechnet werden.
- Es kann im Anwendungsgebiet nur entweder die Vorlesung „*Data Mining für Technische Anwendungen*“ oder die Vorlesung „*Knowledge Discovery*“ angerechnet werden.
- Es kann im Anwendungsgebiet nur entweder die Vorlesung „*Digitale Systeme*“ oder die Vorlesung „*Signalverarbeitung mit Mikroprozessoren I*“ angerechnet werden.

Anwendungsgebiet Knowledge & Data Engineering (Stumme) Seite

Für Basis Anwendungsgebiet 6 CP, für Anwendungsgebiet 12 CP wählen:

| | |
|----------------------------------------|-----|
| Data Mining für Technische Anwendungen | 51 |
| Internet-Suchmaschinen | 84 |
| Knowledge Discovery | 86 |
| Praktikum Internet-Suchmaschinen | 108 |
| Praktikum Knowledge Discovery | 109 |
| Soft Computing | 121 |

Anwendungsgebiet Kommunikation und Verteilte Systeme (Geihs) Seite

Basis Anwendungsgebiet (6 CP nach Wahl)

| | |
|-----------------------------------------------|-----|
| Sicherheit in Kommunikationsnetzen | 118 |
| Techniken und Dienste des Internets | 126 |
| Verteilte Systeme – Architekturen und Dienste | 130 |

Anwendungsgebiet (12 CP nach Wahl)

| | |
|-------------------------------------------------|-----|
| Ausgewählte Kapitel der Kommunikationstechnik | 41 |
| Grundlagen der angewandten Kryptologie | 71 |
| Industrielle Netzwerke | 77 |
| Java Code-Camp Context Awareness I | 85 |
| Programmierung mobiler Geräte (Android und iOS) | 114 |
| Sicherheit in Kommunikationsnetzen | 118 |
| Techniken und Dienste des Internets | 126 |
| Verteilte Systeme – Architekturen und Dienste | 130 |

Anwendungsgebiet Modellierung und Simulation (Linnemann) Seite

Basis Anwendungsgebiet

| | |
|------------------------------------|-----|
| Einführung in Matlab/Simulink | 62 |
| Technische Systeme im Zustandsraum | 128 |

Anwendungsgebiet (12 CP nach Wahl)

| | |
|----------------------------------------------------|----|
| Elektrische und elektronische Systeme im Automobil | 63 |
| Ereignisdiskrete Systeme und Steuerungstheorie | 65 |
| Grundlagen der Regelungstechnik | 74 |

Studierende des Anwendungsgebiets „Modellierung und Simulation“ können alternativ zu den oben angegebenen Veranstaltungen in den Bereich Wahlpflicht „Technische Informatik“ einbringen:

| | |
|------------------------------------------------|----|
| Ereignisdiskrete Systeme und Steuerungstheorie | 65 |
| Grundlagen der Regelungstechnik | 74 |

Anwendungsgebiet Prozessor- und Rechnertechnologie (Börcsök) Seite

Basis Anwendungsgebiet

| | |
|--------------------------------------------------|----|
| Mikroprozessortechnik und eingebettete Systeme 1 | 98 |
|--------------------------------------------------|----|

Anwendungsgebiet (12 CP nach Wahl)

| | |
|----------------------------------------------------------------------|-----|
| Ausgewählte Kapitel der Rechnerarchitektur | 42 |
| Ausgewählte Kapitel der Rechnertechnologie und Mikroprozessortechnik | 43 |
| Digitale Systeme | 55 |
| Echtzeitsysteme | 56 |
| Intelligente Technische Systeme | 83 |
| Mathematische Modelle zur MTTF und PFD Analyse | 94 |
| Mikroprozessortechnik und eingebettete Systeme 2 | 99 |
| Praktikum Intelligente Eingebettete Systeme | 107 |
| Risikobewertung von Rechnerarchitekturen I | 117 |
| Signalverarbeitung mit Mikroprozessoren I | 120 |
| VHDL – Kurs | 132 |
| VHDL – Praktikum | 133 |

Anwendungsgebiet Robotik (Geihs) Seite

Basis Anwendungsgebiet

| | |
|-------------------------|----|
| Autonome Mobile Roboter | 44 |
|-------------------------|----|

Anwendungsgebiet

12 CP nach Wahl mit folgenden Einschränkungen:

- Es kann entweder die Vorlesung „*Computational Intelligence in der Automatisierung*“ oder die Vorlesung „*Soft Computing*“ angerechnet werden.
- Es können maximal 6 Credit Points durch Praktika abgedeckt werden.

| | |
|-----------------------------------------------------------------------------|-----------|
| Computational Intelligence in der Automatisierung oder Soft Computing | 49 121 |
| Einführung in die Aktorik | 57 |
| Grundlagen der Regelungstechnik | 74 |
| Intelligent Humanoid Robots I | 82 |
| Intelligente Technische Systeme | 83 |
| Knowledge Discovery | 86 |
| Mikroprozessortechnik – Labor | 97 |
| Praktikum Intelligente Eingebettete Systeme | 107 |
| Praktikum kooperative verteilte Robotersysteme | 110 |

Anwendungsgebiet Software Tools (Zündorf) Seite

Basis Anwendungsgebiet

| | |
|-------------------------|-----|
| Parallelverarbeitung I | 103 |
| Parallelverarbeitung II | 104 |

Anwendungsgebiet (12 CP nach Wahl)

| | |
|----------------------------------------------------------------------------------------|-----|
| C++ für Fortgeschrittene | 47 |
| Compilerbau und Reverse Engineering | 48 |
| Design Patterns | 53 |
| Einführung in die formale Verifikation | 58 |
| Funktionale Programmierung | 67 |
| Nutzungsorientierte Gestaltung | 102 |
| Seminar in den Fachgebieten Programmiersprachen/–methodik oder Software Engineering | 137 |
| Softwareergonomie | 122 |

Anwendungsgebiet Sozio–technisches Systemdesign (Leimeister) Seite

Für Basis Anwendungsgebiet 6 CP, für Anwendungsgebiet 12 CP wählen:

| | |
|-------------------------------------------------|-----|
| Arbeitswissenschaft | 39 |
| Assistenzsysteme | 40 |
| Betriebliche Anwendung von Internettechnologien | 46 |
| Datenschutzrecht | 52 |
| Informatik und Gesellschaft | 78 |
| Informationswirtschaft | 80 |
| Informationswissenschaften I | 81 |
| Managing IT–enabeld Change | 92 |
| Mensch–Maschine–Systeme 1 | 95 |
| Mensch–Maschine–Systeme 2 | 96 |
| Nutzungsorientierte Gestaltung | 102 |
| Praktikum Mensch–Maschine–Interaktion | 112 |
| Technik– und Produktrecht | 127 |

| Anwendungsgebiet Umweltinformatik (Schaldach) | Seite |
|-------------------------------------------------------------------|--------------|
| <u>Basis Anwendungsgebiet</u> (6 CP nach Wahl) | |
| Einführung in die Umweltinformatik | 60 |
| Einführung in die Umweltwissenschaften | 61 |
| Umweltwissenschaftliche Grundlagen für Ingenieure | 129 |
| <u>Anwendungsgebiet</u> | |
| Geographische Informationssysteme (GIS) | 68 |
| Life Cycle Engineering | 90 |
| Life Cycle Engineering Praktikum | 91 |
| Modellbildung und Simulation: Lokale und regionale Umweltprobleme | 101 |

Pflichtmodule

| | |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Algorithmen und Datenstrukturen |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | AlgoDS |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 2 |
| Modulverantwortliche(r): | Prof. Dr. Claudia Fohry |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor Mathematik Bachelor u. andere |
| Lehrform/SWS: | Vorlesung, Übung jeweils 2SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Empfohlene Voraussetzungen: | Einführung in die Programmierung für Informatik |
| Angestrebte Lernergebnisse | Kenntnis grundlegender Algorithmen und DS der Informatik, Fertigkeiten im Erfassen gegebener Algorithmen, Fertigkeit im Entwickeln eigener Algorithmen und Datenstrukturen, Fertigkeiten in Effizienz- und Korrektheitsanalyse gegebener Algorithmen, vertiefte Fertigkeiten in der Umsetzung von Algorithmen als Programm |
| Inhalt: | Begriffliche Grundlagen zu Algorithmen und Datenstrukturen, Such und Sortierverfahren sowie weitere Grundalgorithmen, Listen und Bäume, Hash-Verfahren, O-Notation, Korrektheit |
| Studien-/Prüfungsleistungen: | SL: Regelmäßige Bearbeitung v. Übungsaufgaben PL: 1 Klausur (90 – 150 Min am Semesterende) |

| | |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Analysis für Informatiker |
| Art des Moduls | Pflichtmodul |
| Studiensemester: | 2 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über eine angemessene mathematische Grundbildung im Bereich der Analysis für Informatiker, die es ihnen erlaubt, mathematischen Argumentationen, wie sie in der Informatik und ihren Anwendungen üblich sind, zu folgen und entsprechende Herleitungen selbst vorzunehmen. |
| Modulverantwortliche/r | Koepf |
| Lehrinhalte | Differential- und Integralrechnung einer Variablen: Folgen, Stetige Funktionen, Umkehrfunktionen, Differenzierbare Funktionen, Integration, Taylorentwicklung |
| Lehr-/ Lernformen (Organisationsform) | 3 SWS Vorlesung, 2 SWS Übung |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | ein Semester im jährlichen Rhythmus |
| Sprache | deutsch |
| Empfohlene Voraussetzungen | Lineare Algebra |
| Studentischer Arbeitsaufwand | 180 Stunden, davon 75 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (60 – 90 min.) |
| Anzahl Credits für das Modul | 6 |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Betriebssysteme |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | BS |
| Studiensemester: | 3 |
| Modulverantwortlicher: | Prof. Dr. K. Geihs |
| Sprache: | Deutsch |
| Lehrform / SWS: | 2 SWS Vorlesung und 2 SWS Übungen |
| Arbeitsaufwand: | 60 Stunden Präsenz 80 Stunden Übungen 20 Stunden Vor- und Nachbereitung von Vorlesung und Übung 20 Stunden Prüfungsvorbereitung Summe: 180 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen der Informatik und Stochastik |
| Angestrebte Lernergebnisse: | <ul style="list-style-type: none"> • Kenntnisse und kritische Beurteilung von Strukturen, Algorithmen der Betriebsmittelverwaltung, Prozesskonzept und -synchronisation, Sicherheitskonzepte • Verstehen von Implementierungsbeispielen in populären Betriebssystemen • Anwendung der Leistungsbewertung von Entwurfsentscheidungen • Einübung der Konzepte mit praktischen Aufgaben |
| Inhalt: | Grundlagen von Rechnerbetriebssystemen: Architekturen, Funktionen, Komponenten, Implementierungsbeispiele |
| Studien-/Prüfungsleistungen: | Klausur (90 - 120 min.) |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Datenbanken |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | DB |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 4 |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Bachelor Informatik Bachelor Elektrotechnik (Wahlpflicht) Bachelor Mathematik (Wahlpflicht) |
| Lehrform/SWS: | Vorlesung 2 SWS, Übungen 2 SWS; 20–30 Teilnehmer, einschließlich Übungen SQL am Rechner |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium, |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Algorithmen und Datenstrukturen |
| Angestrebte Lernergebnisse | Vorteile des Einsatzes von Datenbanken in der Praxis kennen, einfache Anwendungen modellieren, die Grundlagen des Relationenmodells, seine Operationen, funktionale Abhängigkeiten und das Prinzip der Normalisierung verstehen und an Beispieltabellen demonstrieren, die praktische Umsetzung in SQL beherrschen, mittels zweier Basistechniken einfache Operationsfolgen auf Konfliktfreiheit prüfen, die Unterschiede zu anderen Datenmodellen beurteilen können |
| Inhalt: | Schichtenarchitektur ANSI SPARC, ER-Modellierung, das relationale Modell, relationale Algebra, tupelrelationales Kalkül, SQL, funktionale Abhängigkeiten, Normalisierung, Transaktionskonzept, physische Speicherstrukturen, hierarchisches und Netzwerkmodell, OODBMS |
| Studien-/Prüfungsleistungen: | Klausur (90 – 150 Min) |

| | |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Differenzierungsmodul |
| Art des Moduls | Wahlpflichtmodul |
| Modulverantwortliche/r | a) Prof. Dr. Reinhard Hochmuth, Prof. Dr. Wolfram Koepf, Prof. Dr. Arno Linnemann b) gemäß Modulbeschreibung des jeweiligen Bereichs c) Prof. Dr. Reinhard Hochmuth, Prof. Dr. Wolfram Koepf, Prof. Dr. Arno Linnemann d) Dr. Norbert Hundeshagen, Prof. Dr. Martin Lange |
| Studiensemester: | 1 |
| Angestrebte Lernergebnisse | <p>Das Differenzierungsmodul dient</p> <p>a) der Schaffung einer soliden Basis im Bereich mathematischer Rechentechniken sowie ggf. dem Ausgleich von Defiziten und der Auffrischung von Kenntnissen und Fähigkeiten oder b) der Erweiterung der universitären Allgemeinbildung bzw. der Stärkung fachnaher oder fachfremder Kompetenzen oder c) der Vorbereitung der Klausurteilnahme an der jeweils nicht im laufenden Semester als Vorlesung angebotenen Mathematikveranstaltung (Lineare Algebra bzw. Analysis). d) dem Erfassen und Verstehen von formalen Beschreibungen für Systeme und Methoden der Informatik, wie z.B. Programme, Daten, Beweise, Rekursion usw.. Das Ziel dieser Veranstaltung ist es, die mathematischen Grundlagen zu legen, die es den Studierenden in späteren Modulen ermöglichen, mit formalen, mathematischen Beschreibungen umzugehen.</p> <p>Angestrebte Kompetenzen zu a): Die Studierenden können</p> <ul style="list-style-type: none"> – Funktionen in Bezug auf elementare Eigenschaften untersuchen, Rechengesetze auf lineare, quadratische und Potenz-Funktionen anwenden, – mit Polynomen, Exponentialfunktionen, Logarithmusfunktionen, trigonometrischen Funktionen und einfachen rationalen Funktionen umgehen und rechnen, – das Änderungsverhalten von Funktionen analytisch beschreiben und interpretieren, – Polynome, Exponentialfunktionen, Logarithmusfunktionen, trigonometrische Funktionen und einfache rationale Funktionen ableiten, – Ableitungsregeln (Produkt, Quotienten, Verknüpfung) anwenden, – Extremwertaufgaben lösen, – Kurvendiskussionen in Bezug auf lokale und globale Eigenschaften durchführen und interpretieren, – das bestimmte Integral als Flächeninhalt deuten, – den Zusammenhang zwischen Ableitung und Integral ausnutzen und interpretieren, |

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> - das unbestimmte Integral von Polynomen, Exponentialfunktionen, Logarithmusfunktionen, trigonometrischen Funktionen und einfachen rationalen Funktionen bestimmen, - Integrationsregeln (partielle Integration, Substitution) anwenden, - die Partialbruchzerlegung zur Berechnung von Integralen anwenden, - lineare Gleichungssysteme interpretieren und mit Hilfe des Eliminationsverfahrens lösen, - Lösungsmengen linearer Gleichungssysteme bestimmen und interpretieren, - die bildliche Darstellung von Aufgaben in der Ebene und im Raum ausnutzen und interpretieren,- mit Vektoren, Geraden und Ebenen arbeiten, - Winkel, Längen und Abstände bestimmen und graphisch interpretieren. <p>Die angestrebten Lernergebnisse zu b) ergeben sich aus der Modulbeschreibung des gewählten Bereichs.</p> <p>Angestrebte Kompetenzen zu c): Die Studierenden verfügen über die mathematischen Grundlagen im Bereich der Linearen Algebra bzw. der Analysis. Fast Track zur Linearen Algebra: Die Studierenden können</p> <ul style="list-style-type: none"> - lineare Gleichungssysteme lösen, - mit Matrizen umgehen, - Eigenwerte und Eigenvektoren berechnen, - mathematische Probleme aus diesem Bereich selbständig lösen. <p>Fast Track zur Analysis: Die Studierenden können</p> <ul style="list-style-type: none"> - Eigenschaften reeller Funktionen bestimmen, - differenzieren und integrieren, - mit Reihen umgehen, - mathematische Probleme aus diesem Bereich selbständig lösen. <p>Angestrebte Kompetenzen zu d):</p> <ul style="list-style-type: none"> - Erfassung und Umsetzung von formalen Beschreibungen für Systeme und Methoden der Informatik, wie z.B. Programme, Daten, Beweise, Rekursion - Umgang mit formalen, mathematischen Beschreibungen |
| Lehrinhalte | <p>a) Mathematischer Brückenkurs (für Studierende, die den Mathematiktest gemäß § 7 nicht bestanden haben und Studierende, die den Mathematiktest bestanden haben, ihre Mathematikkenntnisse aber weiter vertiefen wollen):</p> <p><u>1. Funktionsbegriff und elementare Funktionen</u> Rechengesetze, Potenzen, lineare und quadratische Funktionen</p> |

| | |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p><u>2. Höhere Funktion</u> Polynome, Exponentialfunktion, Logarithmus, trigonometrische Funktionen</p> <p><u>3. Differentialrechnung</u> Grenzwerte von Folgen und Funktionen, Ableitung an einer Stelle, Ableitungsfunktion, Produkt-, Quotienten- und Kettenregel, Ableitung höherer Funktionen, Extremwertaufgaben</p> <p><u>4. Integralrechnung</u> Flächen, bestimmte Integrale, unbestimmte Integrale, wichtige Stammfunktionen, partielle Integration, Substitution, Anwendung der Partialbruchzerlegung</p> <p><u>5. Lineare Algebra</u> Lineare Gleichungssysteme, Vektoren, Geraden, Ebenen, Winkel, Abstände.</p> <p>b) Module oder Lehrveranstaltungen aus anderen Bereichen, z.B. dem Bereich der Schlüsselkompetenzen, der Informatik, der Naturwissenschaften oder einer anderen Ingenieur-Disziplin (für Studierende, die den Mathematiktest gemäß § 7 bestanden haben): Lehrinhalte gemäß der Modulbeschreibung des jeweiligen Bereichs.</p> <p>c) Fast Track: Mathematik – Begabtenförderung Die Lehrinhalte sollen von den Studierenden unter Anleitung der/des Dozent/in weitestgehend selbständig erarbeitet werden. Im Sommersemester findet der Fast Track zur Linearen Algebra statt und richtet sich im Wesentlichen nach dem Buch "Höhere Mathematik 1" von W. Strampp: Vektorrechnung, Vektorräume, komplexe Zahlen, Matrizen, lineare Gleichungssysteme, Determinanten, Eigenwerte, Eigenvektoren. Im Wintersemester findet der Fast Track zur Analysis statt und richtet sich im Wesentlichen nach dem Buch "Höhere Mathematik 2" von W. Strampp: Reelle Zahlen, Folgen, Funktionen, Stetigkeit, Differential- und Integralrechnung einer Variablen, Taylorentwicklung, Reihen, Mehrdimensionale Differential- und Integralrechnung. Der Fast Track zur Analysis für Informatiker behandelt keine mehrdimensionale Differential- und Integralrechnung.</p> <p>d) Vorbereitungskurs Theoretische Informatik und Diskrete Strukturen</p> <ul style="list-style-type: none"> - Sprache der Mathematik (Mengen, Funktionen, Relationen, Variablen, Gleichungen, Wahrheit, ...) |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> - Sprache der Informatik (Syntax vs. Semantik, Ausdrücke vs. Werte, ...) - allgemeine Beweistechniken (Folgerung, Induktion, Widerspruch, Kombinatorik, ...) - Anwendung obiger Punkte auf formale Beschreibungen wie z.B.: Terme, Programme, ... |
| Lehr-/ Lernformen (Organisationsform) | <ul style="list-style-type: none"> a) Kurs, 4 SWS b) gemäß Modulbeschreibung des jeweiligen Bereichs c) 1 SWS Tutorium, 2 SWS Übung d) in interaktiver Form, selbständig, anhand vorgegebener Materialien auf Themenbereiche vorbereiten und aktiv an den Präsenzstunden teilnehmen; 2 SWS |
| Verwendbarkeit des Moduls | B. Sc. Informatik |
| Dauer und Häufigkeit des Angebotes | ein Semester, Angebote in jedem Semester |
| Sprache | Variierend |
| Empfohlene Voraussetzungen | <ul style="list-style-type: none"> b) bestandener Mathematiktest nach § 7 c) Überdurchschnittliche Leistungen im Mathematiktest d) bestandener Mathematiktest nach § 7 |
| Studentischer Arbeitsaufwand | <ul style="list-style-type: none"> a) 60 Stunden Kursteilnahme, 30 Stunden Selbststudium b) gemäß Modulbeschreibung des jeweiligen Bereichs c) 90h: 45h Präsenz, 45h Selbststudium d) 2 SWS Präsenz, Selbststudium |
| Studien- und Prüfungsleistung | <ul style="list-style-type: none"> a) Studienleistungen: Teilnahme an Präsenzveranstaltungen, regelmäßige Bearbeitung von Übungsaufgaben, eigenständige Beseitigung individueller Defizite in Selbstlernphasen, abschließende Klausur (Dauer 45 bis 90 Minuten). Eine nicht bestandene abschließende Klausur kann beliebig oft wiederholt werden. b) Modulprüfungsleistung oder Studienleistung nach Vorgabe des gewählten Bereiches. Bei endgültigem Nichtbestehen kann ein weiteres Modul gewählt werden. Die Note gem. a) und b) geht nicht in die Bachelornote ein. c) Studienleistung: Selbstlernphasen zum Aufarbeiten des Lehrstoffes mithilfe der zur Verfügung gestellten Lernhilfen, regelmäßiges Vorrechnen und Abgabe von Übungsaufgaben d) Klausur am Ende des Semesters sowie aktive Beteiligung an der Veranstaltung (Kurzvorträge, Vorrechnen von Übungsaufgaben). |
| Anzahl Credits für das Modul | <p>3</p> <p>Zusätzlich bei c) Zulassung zur jeweiligen Klausur (Lineare Algebra oder Analysis)</p> |

| | |
|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Digitale Logik |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Untertitel | Digitaltechnik I |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 1 |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Elektrotechnik Bachelor Informatik Bachelor Mechatronik Bachelor |
| Lehrform/SWS: | 3 SWS: 2 SWS Vorlesung 1 SWS Übung |
| Arbeitsaufwand: | 120 h: 45 h Präsenzzeit 75 h Selbststudium |
| Kreditpunkte: | 4 |
| Angestrebte Lernergebnisse | Verständnis der grundlegenden Funktionsweise digitaler Schaltungen und deren Anwendung. Studenten sollen in die Lage versetzt werden, einfache Digitalisierungen zu planen, zu optimieren und zu analysieren. |
| Inhalt: | Zahlendarstellung und Codes, Boolesche Algebra, Entwurf und Vereinfachung von Schaltnetzen, Analyse und Synthese von Schaltwerken, Steuerwerksentwurf, Mikroprogrammsteuerung. |
| Klausur, Studienleistungen (b/nb): Übungsaufgaben | Prüfungsleistung: Klausur (ca. 90 min.) Studienleistungen (b/nb): Übungsaufgaben |

| | |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Diskrete Strukturen I |
| Art des Moduls | Pflichtmodul |
| Modulverantwortliche/r | Koepf |
| Studiensemester: | 2 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über eine angemessene mathematische Grundbildung im Bereich der Diskreten Strukturen, die es ihnen erlaubt, mathematischen Argumentationen, wie sie in der Informatik und ihren Anwendungen üblich sind, zu folgen und entsprechende Herleitungen selbst vorzunehmen. |
| Lehrinhalte | Kombinatorik; diskrete Wahrscheinlichkeitstheorie; Elemente der Statistik; Rekursionsgleichungen und erzeugende Funktionen |
| Lehr-/ Lernformen (Organisationsform) | 2 SWS Vorlesung, 2 SWS Übung |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | zwei Semester im jährlichen Rhythmus |
| Sprache | deutsch |
| Empfohlene Voraussetzungen | Lineare Algebra |
| Studentischer Arbeitsaufwand | 180 Stunden, davon 60 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (90 – 120 min.) am Semesterende |
| Anzahl Credits für das Modul | 6 |

| | |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Diskrete Strukturen II |
| Art des Moduls | Pflichtmodul |
| Modulverantwortliche/r | Koepf |
| Studiensemester: | 3 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über eine angemessene mathematische Grundbildung im Bereich der Diskreten Strukturen, die es ihnen erlaubt, mathematischen Argumentationen, wie sie in der Informatik und ihren Anwendungen üblich sind, zu folgen und entsprechende Herleitungen selbst vorzunehmen. |
| Lehrinhalte | Algebra und Arithmetik; Elemente der Kryptographie; Graphentheorie; Boolesche Algebra |
| Lehr-/ Lernformen (Organisationsform) | 2 SWS Vorlesung, 2 SWS Übung |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | zwei Semester im jährlichen Rhythmus |
| Sprache | deutsch |
| Empfohlene Voraussetzungen | Lineare Algebra |
| Studentischer Arbeitsaufwand | 180 Stunden, davon 60 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (90 – 120 min.) am Semesterende. |
| Anzahl Credits für das Modul | 6 |

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in C |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 1 |
| Modulverantwortliche(r): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | 1 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden, davon 30 Stunden Präsenzzeit |
| Kreditpunkte: | 2 |
| Angestrebte Lernergebnisse | Erstellen einfacher Programme in der Programmiersprache C |
| Inhalt: | Grundlegendes zum Programmieren in C, Datentypen, Pointer, Steuerung des Programmflusses, Präprozessor, Operatoren, Funktionen, Rückgabe von Werten, Bibliotheken |
| Studien-/Prüfungsleistungen: | Prüfung in der Regel als E-Klausur (60 min.) am Semesterende. |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Künstliche Intelligenz |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | EKI |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 5 |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Dozent(in): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor, Mathematik (Bachelor) |
| Lehrform/SWS: | Vorlesung + Übung, je 1 SWS |
| Arbeitsaufwand: | 90 Stunden, davon 30 Stunden Präsenz |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Algorithmen und Datenstrukturen |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über eine angemessene Grundbildung im Bereich der Künstlichen Intelligenz, die es ihnen erlaubt, Methoden der KI für den jeweiligen Anwendungskontext auszuwählen und umzusetzen. Die Studierenden sind in der Lage, diese Methoden zu Ansätzen angrenzender Forschungsbereiche in Bezug zu setzen. |
| Inhalt: | Die Veranstaltung zeigt das Spektrum von Methoden der Künstlichen Intelligenz für die Verarbeitung von Wissen mit dem Rechner auf. Die Vorlesung gibt einen Überblick über verschiedene Gebiete der Wissensrepräsentation und führt hin zu aktuellen Einsatzszenarien wie der Erweiterung des World Wide Web hin zu einem Semantic Web. Ziel ist insbesondere der effiziente Umgang mit Wissen in Internet und Intranet. Themen: Problemlösemethoden, Wissensrepräsentation, Inferenz, Unsicherheit, Ontologien, Semantic Web, XML, RDF, OWL, Social Bookmark Systems, Folksonomies, Anwendungen. |
| Studien-/Prüfungsleistungen: | Klausur (60 – 120 min.) oder mündliche Prüfung (20 – 40 min.) |
| Medienformen: | Beamer, Skript, Tafel |

| | |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Programmierung für Informatik |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | Einfprog |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 1 |
| Modulverantwortliche(r): | Prof. Dr. Claudia Fohry |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mathematik (Bachelor) u. andere |
| Lehrform/SWS: | Vorlesung, Übung jeweils 2SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Angestrebte Lernergebnisse | Gründliche Kenntnisse einer Programmiersprache, Verständnis für Abläufe im Rechner bei Programmausführung, Verstehen grundlegender Programmierkonzepte (z.B. Objektorientierung), gute Fertigkeiten bei Entwicklung prozeduraler Programme bis etwa 100 Zeilen, Fertigkeiten in objektorientierter Programmierung |
| Inhalt: | Grundlagen in einer aktuellen Programmiersprache (z.B. Java): Variablen, Ausdrücke, Kontrollstrukturen, Methoden, Klassen, Vererbung, Bibliotheken |
| Studien-/Prüfungsleistungen: | SL: Regelmäßige Bearbeitung v. Übungsaufgaben PL: 1 Klausur (80 – 120 Min am Semesterende) |

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Elektrotechnik für Informatiker |
| ggf. Modulniveau | Pflichtmodul |
| Studiensemester: | 1 |
| Modulverantwortlicher: | Prof. B. Witzigmann |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik |
| Lehrform / SWS: | Vorlesung / 2 SWS Übung / 1 SWS Tutorium / 2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung, 1 SWS Übung 2 SWS Tutorium Selbststudium: 3 SWS, Prüfungsvorbereitung: 40 h |
| Kreditpunkte: | 5 |
| Empfohlene Voraussetzungen: | Gute Kenntnisse der Mathematik der Oberstufe, Grundbegriffe der Differential- und Integralrechnung, Algebra |
| Angestrebte Lernergebnisse: | <ul style="list-style-type: none"> • Grundlegende Kenntnisse der physikalischen und technischen Zusammenhänge im Umfeld der Elektrotechnik • Kenntnisse und Fertigkeiten in der Anwendung grundlegender Verfahren zur Berechnung von Gleichstromnetzwerken • Fertigkeiten in der Anwendung algebraischer Techniken auf die Grundgleichungen der Elektrotechnik |
| Inhalt: | Einheiten und Gleichungen Grundlegende Begriffe Berechnung von Strömen und Spannungen in elektrischen Gleichstromnetzwerken Elektrostatische Felder Stationäre Magnetfelder Zeitlich veränderliche magnetische Felder |
| Studien-/Prüfungsleistungen: | Regelmäßiges Bearbeiten von Übungs- und Tutoriumsaufgaben/ Klausur (90–150 min.) |

| | |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Grundwissen der Elektronik |
| Art des Moduls | Pflichtmodul |
| Anzahl Credits für das Modul | 3 |
| Studiensemester: | 4 |
| Modulverantwortliche/r | Prof. Dr. rer. nat. H. Hillmer |
| Angestrebte Lernergebnisse | Die Studierenden haben ein überblicksmäßiges Wissen zur Elektrotechnik, das eine spätere Einarbeitung in technische Anwendungen in Studium und Berufsleben erleichtert. Sie verfügen über eine angemessene Grundbildung zur Elektronik, die es ihnen erlaubt, den technischen Hintergrund von Informatik-Systemen zu erfassen und zu bewerten sowie selbst entsprechende Entwicklungen vorzunehmen. |
| Lehrinhalte | Halbleiter, elektronische Bauelemente, integrierte Schaltungen, verschiedene Halbleiterspeicher |
| Lehr-/ Lernformen (Organisationsform) | 2 SWS Vorlesung |
| Verwendbarkeit des Moduls | Mathematik Bachelor, Informatik Bachelor, Wirtschaftsingenieurwesen Bachelor |
| Dauer und Häufigkeit des Angebotes | zwei Semester im jährlichen Rhythmus |
| Sprache | Deutsch |
| Voraussetzungen laut Prüfungsordnung | Keine |
| Empfohlene Voraussetzungen | keine |
| Studentischer Arbeitsaufwand | 95 Stunden, davon 30 Stunden Präsenz |
| Studien- und Prüfungsleistung | Klausur (ca. 60 min.) am Semesterende |

| | |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Lineare Algebra |
| Art des Moduls | Pflichtmodul |
| Modulverantwortliche/r | Koepf |
| Studiensemester: | 1 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über eine angemessene mathematische Grundbildung im Bereich der Linearen Algebra, die es ihnen erlaubt, mathematischen Argumentationen, wie sie in der Informatik und ihren Anwendungen üblich sind, zu folgen und entsprechende Herleitungen selbst vorzunehmen. |
| Lehrinhalte | Reelle und komplexe Zahlen, Vektorrechnung, Vektorräume, Matrizen, Lineare Gleichungssysteme, Determinanten, Eigenwerte |
| Lehr-/ Lernformen (Organisationsform) | 4 SWS Vorlesung, 2 SWS Übung |
| Verwendbarkeit des Moduls | Elektrotechnik Bachelor, Mechatronik Bachelor, Informatik Bachelor, Wirtschaftsingenieur (Fachrichtung Elektrotechnik) Bachelor |
| Dauer und Häufigkeit des Angebotes | ein Semester im jährlichen Rhythmus |
| Sprache | deutsch |
| Empfohlene Voraussetzungen | Mathematischer Vorkurs dringend empfohlen |
| Studentischer Arbeitsaufwand | 210 Stunden, davon 90 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (90 – 120 min.) |
| Anzahl Credits für das Modul | 7 |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Programmiermethodik |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | PM |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 3 |
| Modulverantwortliche(r): | Prof. Dr. Albert Zündorf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatiker |
| Lehrform/SWS: | Vorlesung, 4 SWS, 20–30 Teilnehmer |
| Arbeitsaufwand: | Stunden Präsenz Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Einführung in die Programmierung für Informatik |
| Angestrebte Lernergebnisse | Die Studierenden können eine Problemstellung mit Hilfe von Szenarien analysieren, Objektdiagramme entwerfen und daraus Klassendiagramme ableiten. Die Studierenden können aus diesem Design eine Implementierung ableiten und diese Implementierung durch systematische Tests validieren. |
| Inhalt: | Einfache Vorgehensweise, Anforderungsmodellierung (Usecases), Objektorientierte Modellierung, Analyse (Szenariodiagramme), Ableitung des Designs (Klassendiagramme, Statecharts), systematische Implementierung |
| Studien-/Prüfungsleistungen: | Hausaufgaben und Klausur (100 – 140 min.) |

| | |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Rechnerarchitektur |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 2 |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung/2 SWS Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 180 Stunden |
| Kreditpunkte: | 6 |
| Empfohlene Voraussetzungen: | Digitale Logik, Programmierkenntnisse |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> • Kenntnis des grundsätzlichen Aufbaus unterschiedlicher Architekturen und deren Merkmale sowie des Aufbaus und der Wirkungsweise von Rechnerkomponenten • Fertigkeiten im Entwurf von Rechnerarchitekturen (Modellierung etc.) |
| Inhalt: | Grundlagen der Informationsdarstellung in Rechenanlagen, Codierung, Bewertungskriterien von Rechnerarchitekturen, Grundsätzlicher Aufbau unterschiedlicher Architekturen und deren Merkmale, Aufbau und Wirkungsweise von Rechnerkomponenten (ALU, AKKU, Systembus, etc.), Automaten, Aufbau einer Einfacharchitektur. |
| Studien-/Prüfungsleistungen: | Klausur (60 – 120 min.) oder mündliche Prüfung (20 – 40 min.) oder Hausarbeit (25 – 30 Seiten) |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Rechnernetze |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 3 |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Elektrotechnik (Bachelor), Informatik (Bachelor) |
| Lehrform/SWS: | Vorlesung, Übung, 4 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Erfolgreiche Teilnahme an den ersten 2 Semestern eines technischen (Informatik/ E-Technik) Studiums |
| Angestrebte Lernergebnisse | Kenntnis grundlegender Techniken und Prinzipien der Kommunikationsnetze und Anwendungen; Berechnungen zu Mindeststrahlengrößen, Quell-, Kanal- und Leitungskodierung, Adressierung, Paketanalyse |
| Inhalt: | Beispiele für Inhalte sind: <ul style="list-style-type: none"> • OSI 7 Schicht Kommunikationsmodell (physikalische, logische, Peer-to-peer, SAP), • Layer 1: versch. Übertragungsmedien wie CAT5, optische Fasern, Funk, Dispersion, Dämpfung, Stecker • Layer 2: MAC, LLC, NIC, Hardwareadressierung • Layer 3: ISDN, IP, Routing • Layer 4: UDP, TCP • Layer 5-7 Anwendungen wie: http, email, WWW, Telnet • evtl. aktuelle Vertiefungen wie: DSL, W-LAN, VoIP, "Security" |
| Studien-/Prüfungsleistungen: | Klausur (90 - 120 min.) oder mündliche Prüfung (20 - 40 min.) |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Softwaretechnik I |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | SE |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | 4 |
| Modulverantwortliche(r): | Prof. Dr. Albert Zündorf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik |
| Lehrform/SWS: | Vorlesung, 4 SWS, 60 Teilnehmer |
| Arbeitsaufwand: | Stunden Präsenz Stunden Eigenstudium |
| Kreditpunkte: | 9 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Einführung in die Programmierung für Informatik, Programmiermethodik |
| Angestrebte Lernergebnisse | Die Studierenden können für ein Softwareprojekt geeignete Methoden und Werkzeuge auswählen und an das Projekt anpassen. Sie können in einem Teamprojekt mit vorgegebenen Methoden und Werkzeugen teilnehmen und die einzelnen Schritte ausführen. |
| Inhalt: | Moderne Vorgehensmodelle (RUP, XP), Qualitätssicherung (Testverfahren, Reviews, etc.), Projektplanung und -verfolgung (Schätzverfahren, Software-Projektmanagement). |
| Studien-/Prüfungsleistungen: | Projektarbeit |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Systemprogrammierung |
| ggf. Modulniveau | Pflichtmodul |
| ggf. Kürzel | |
| Studiensemester: | 4 |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Maschinenbau (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung / 2 SWS Übung / 1 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 1 SWS Übung (15 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 5 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, Betriebssysteme, Grundlagen der Mathematik (Stochastik) |
| Angestrebte Lernergebnisse | Kenntnis des Aufbaus und Zusammenspiels von Systemprogrammen und deren Bewertungsmöglichkeiten; Grundlagen der Systemprogrammierung. |
| Inhalt: | Systemprogrammen und deren Bewertungsmöglichkeiten. Grundlagen der Systemprogrammierung, Bewertungskriterien von Systemsoftwarekomponenten, Modelle der Systemsoftwareentwicklung. Beispielanwendungen aus verschiedenen Applikationen |
| Studien-/Prüfungsleistungen: | Studienleistung: Programmierung eines treibers Prüfungsleistung: Klausur (60–120 Minuten), mündliche Prüfung (20–40 Minuten), Hausarbeit (25–30 Seiten) oder Vortrag (30–45 Minuten) |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Theoretische Informatik – Berechenbarkeit und Formale Sprachen |
| Art des Moduls | Pflichtmodul |
| Modulverantwortlicher | Otto |
| Studiensemester: | 2 |
| Angestrebte Lernergebnisse | Die Studierenden verstehen die Grundlagen der Formalen Sprachen, der Berechenbarkeit und Komplexität. Sie verfügen über die Fähigkeit zur Anwendung in der Informatik. |
| Lerninhalte | Endliche Automaten und reguläre Sprachen, Kellerautomaten und kontextfreie Sprachen, Turing-Maschinen, Berechenbarkeit, rekursive Aufzählbarkeit, Church'sche These, Unentscheidbarkeit |
| Lehr-/Lernformen (Organisationsform) | 3 SWS Vorlesung, 1 SWS Übung |
| Verwendbarkeit des Moduls | Mathematik Bachelor, Informatik Bachelor, Wahlmodul in weiteren Studiengängen |
| Dauer und Häufigkeit des Angebotes | 1 x pro Studienjahr |
| Sprache | deutsch |
| Empfohlene Voraussetzungen | Diskrete Strukturen I |
| Studentischer Arbeitsaufwand | 180 Stunden, davon 60 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (90 – 150 min.) |
| Anzahl Credits für das Modul | 6 |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Modulname | Theoretische Informatik – Logik |
| Art des Moduls | Pflichtmodul |
| Modulverantwortlicher | Lange |
| Studiensemester: | 3 |
| Angestrebte Lernergebnisse | Die Studierenden verstehen die Grundlagen der Logik. Sie verfügen über die Fähigkeit zur Anwendung in der Informatik. |
| Lerninhalte | Aussagenlogik, Prädikatenlogik erster Stufe, Modelle, Resolution, Korrektheit von Programmen, Logikprogrammierung |
| Lehr-/Lernformen (Organisationsform) | 3 SWS Vorlesung, 1 SWS Übung |
| Verwendbarkeit des Moduls | Mathematik Bachelor, Informatik Bachelor, Wahlmodul in weiteren Studiengängen |
| Dauer und Häufigkeit des Angebotes | 1 x pro Studienjahr |
| Sprache | deutsch |
| Studentischer Arbeitsaufwand | 180 Stunden, davon 60 Stunden Präsenz |
| Studien- und Prüfungsleistung | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Klausur (90 – 150 min.) |
| Anzahl Credits für das Modul | 6 |

Wahlpflichtmodule

Weitere Wahlpflichtmodule des aktuellen Lehrangebots können nach Genehmigung durch den Prüfungsausschuss belegt werden.

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | 3D Modellierung |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | 3D Mod. |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Vorlesung, 2 SWS, Übung 2 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | keine |
| Angestrebte Lernergebnisse | Erlernen der wichtigsten Konzepte des Bereichs 3D Modellierung für die Erstellung von Avataren in Serious Games. |
| Inhalt: | Grundkonzepte der graphischen 3D Modellierung – Elementare Einführung in 3D Studio Max – Modellierkonzepte für Avatare – Transfer der Avatare in Unity |
| Studien-/Prüfungsleistungen: | Studienleistungen werden vorausgesetzt, siehe moodle Prüfungsleistung: Projekt und Hausarbeit |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Advanced Game Programming |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Praktikum, 2 SWS |
| Arbeitsaufwand: | 30 Stunden Präsenz 60 Stunden Eigenstudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Praktikum 3D Game Programming |
| Angestrebte Lernergebnisse | Fundierte Kenntnisse über aktuelle Game-Programming Techniken. Vertiefte Kenntnisse der allgemeinen Funktionsweise von aktuell verfügbaren Game-Engines und die Erweiterung dieser mit State-of-the Art Entwicklungstools. Fertigkeit ein Spiel zu planen, zu entwickeln und zu testen. |
| Inhalt: | Das Praktikum erweitert und vertieft die im ersten Teil des Praktikums entwickelten Ideen. Es werden spezielle Tools zur Entwicklung von Spielen eingesetzt. Zusätzlich werden Online-Kollaborationstools genutzt, um eine Teamkommunikation zu gewährleisten. Typische Spielentwicklungsrollen werden von den Studierenden eingenommen (3D Content Artist, Projektleiter, Animator, ...). Zusätzlich wird die Anwendung mit externen Testern getestet und das Feedback ausgewertet. Geplante Themen sind <ul style="list-style-type: none"> - Edutainment - Requirementmanagement - GUI-Programmierung - Team-Entwicklung - Testen |
| Studien-/Prüfungsleistungen: | PL: Hausarbeit |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Algebra |
| ggf. Modulniveau | Anwendungsgebiet Computational Mathematics |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Dozenten des FB 10 Mathematik und Naturwissenschaften |
| Sprache: | Meist Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mathematik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 45 Stunden Präsenz 105 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker, Grundlagen der Algebra und Computeralgebra |
| Angestrebte Lernergebnisse | Dieses Modul bietet die Gelegenheit, sich grundsätzlich und systematisch mit Abstraktion, Modellbildung und formalen Techniken zu befassen. Dabei soll der Erkenntniswert abstrakten Denkens demonstriert und die Nützlichkeit theoretischer Modelle zur Behandlung konkreter Probleme aufgezeigt und die dazu nötigen Fähigkeiten vermittelt werden. |
| Inhalt: | Mögliche Themen sind Galoistheorie I, Computeralgebra I, Kryptographie I oder Kodierungstheorie I. Welche der jeweils aktuellen Lehrveranstaltungen diesem Modul zugeordnet sind, wird zusammen mit einer detaillierten Inhaltsbeschreibung im Vorlesungsverzeichnis des FB 10 (Modul B6) ausgewiesen. |
| Studien-/Prüfungsleistungen: | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben; der Dozent kann für einzelne Lehrveranstaltungen die Bearbeitung der Übungsaufgaben auch ganz oder teilweise durch Kurzreferate oder Hausarbeiten ersetzen. Prüfungsleistung: Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Applikationsentwicklung für Tablet-Computer |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Praktikum 4 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Vorbereitungszeit |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Gute Programmierkenntnisse in C/C++ |
| Angestrebte Lernergebnisse | <p>Die/der Lernende</p> <ul style="list-style-type: none"> - Aufbau und Ablauf einer iPad-App erläutern, - wesentliche Konzepte beim Einsatz von Objective-C wiedergeben - typische Entwurfsmuster zur App-Erstellung anwenden, - Schlüsselprobleme einer geplanten Benutzerschnittstelle analysieren, - Problemlösungen in Form von Programmfunktionen entwickeln - netzwerkbasierte Softwarefunktionen implementieren - selbst implementierte Programmfunktionen erläutern und vortragen. |
| Inhalt: | Ein Programm für einen Tablet-Computer (iPad-App) mit hohem Anteil an Benutzerinteraktion soll geplant und implementiert werden. Schwerpunkt ist das Einüben der Verwendung einer graphischen Benutzerinteraktion an einem konkreten Programmbeispiel. Ein weiterer Bestandteil ist die Netzwerkanbindung der App an einen Serverprozess und Test und Inbetriebnahme der Software. |
| Studien-/Prüfungsleistungen: | Referat/Präsentation, Abgabe des erstellten Codes, Teamarbeit und Vorführung der Ergebnisse |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Arbeitswissenschaft |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | AW |
| Studiensemester: | ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Ludger Schmidt |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 4SWS: 2 SWS Vorlesung 1 SWS Übung 1 SWS Seminar |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden (Vorlesung) 15 Stunden (Übung) 15 Stunden (Seminar) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | Die Studierenden haben ein breites und integriertes Wissen arbeitswissenschaftlicher Grundlagen und sind in der Lage ihr Wissen selbstständig zu vertiefen. |
| Inhalt: | Einführung und Belastungs-Beanspruchungs-Konzept Betriebsorganisation Arbeitsorganisation Modellierung und Optimierung von Arbeitsprozessen Zeitstrukturanalyse und experimentelle Zeitermittlungsmethoden Rechnerische Zeitermittlungsmethoden Entgelt und Motivation Arbeitsschutz und sicherheitstechnische Arbeitsgestaltung Arbeitsumgebungsfaktoren Arbeitsplatzgestaltung in der Produktion |
| Studien-/Prüfungsleistungen: | Vorlesung: Schriftliche (90 min.) oder mündliche Prüfung (20 min.) (nach Teilnehmerzahl) Seminar: Seminarvortrag oder Hausarbeit |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Assistenzsysteme |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | AS |
| Studiensemester: | ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Ludger Schmidt |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 3 SWS: 2 SWS Vorlesung 1 SWS Übung |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden (Vorlesung) 15 Stunden (Übung) Selbststudium: 75 Stunden |
| Kreditpunkte: | 4 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Mensch-Maschine-Systeme 1 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über Kenntnisse auf verschiedenen Anwendungsgebieten der Mensch-Maschine-Systeme und über die Möglichkeiten, den Menschen bei seiner Tätigkeit zu unterstützen. Sie können die Grenzen und Risiken solcher Systeme erkennen. |
| Inhalt: | Einführung und konzeptionelle Grundlagen Technische Grundlagen Fahrerassistenz Navigationsassistenz Assistenz bei der Flugführung Prozessüberwachung Teleoperationsunterstützung Hilfesysteme in PC-Anwendungen Mobile Assistenz Ambient Assisted Living Smart Home Patientenüberwachung in der Intensivmedizin |
| Studien-/Prüfungsleistungen: | mündliche Prüfung (20 min.) oder Seminarvortrag |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Ausgewählte Kapitel der Kommunikationstechnik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor) |
| Lehrform/SWS: | Vorlesung, Übung, 4 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Erfolgreiche Teilnahme an „Rechnernetze“ |
| Angestrebte Lernergebnisse | Die Studierenden erhalten exemplarisch Einblick in ausgewählte Themen aus dem Gebiet der Kommunikationstechnik. Die Veranstaltung soll die Studierenden in die Lage versetzen, eigenständig Aspekte der Kommunikationstechnik zu erarbeiten und erläutern zu können. Die Veranstaltung bereitet die Studenten auch auf eine mögliche Bachelor- oder Masterarbeit vor. |
| Inhalt: | Ausgewählte Themen auf dem Gebiet der Kommunikationstechnik |
| Studien-/Prüfungsleistungen: | Klausur, mündliche Prüfung, Vortrag oder Ausarbeitung |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Ausgewählte Kapitel der Rechnerarchitektur |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechnertechnologie |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung/2 SWS Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Programmierkenntnisse, Mikroprozessortechnik I und Mikroprozessortechnik II |
| Angestrebte Lernergebnisse | Vertiefte Kenntnisse der Prozessorarchitektur, VHDL Design, Implementierung von einfachen Architekturen |
| Inhalt: | VHDL-Design, Funktionsweise von Rechnerarchitekturen. Aufbau und Implementierung von einfacheren Rechnerarchitekturen in VHDL. |
| Studien-/Prüfungsleistungen: | schriftliche Prüfung, mündliche Prüfung, Hausarbeit |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Ausgewählte Kapitel der Rechner-technologie und Mikro- prozessortechnik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechner-technologie |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor |
| Lehrform/SWS: | 4SWS: 2SWS Vorlesung 2SWS Übung |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Rechnerarchitektur |
| Angestrebte Lernergebnisse | Der/die Lernende kann: - Kenntnisse über Rechner-technologien vertiefen, - Kenntnisse über Rechnerarchitekturen der vertiefen, - Kenntnisse über Chip-Entwurf erwerben, - Kenntnisse über Testbarkeit von Chip-Entwürfen erwerben. |
| Inhalt: | Die Vorlesung beschäftigt sich mit verschiedenen Themen aus dem Bereich „Rechner-technologie und Mikro- prozessortechnik“. Es gibt Einblicke in den Entwurf von Integrierten Schaltungen, Chip-Design, rekonfigurierbaren Systemen, Fehlertoleranz und Fehlerbehandlung bei Rechnern, sowie in den Einsatz von eingebetteten Systemen bei automotiven Anwendungen, etc. |
| Studien-/Prüfungsleistungen: | Je nach Teilnehmerzahl: Klausur 60 - 180 Minuten oder mündliche Prüfung 20 - 40 Minuten |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Autonome Mobile Roboter |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Basis Anwendungsgebiet Robotik |
| ggf. Kürzel | AMR |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Kurt Geihs |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Vorlesung 2 SWS, Übung 2 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | Kennenlernen einer Roboterarchitektur, Programmierung der Roboterplattform, Sensordatenverarbeitung, Entscheidungsfindung in Robotersystemen, Steuerung der Aktorik, Einblicke in verteilte kooperative Robotersysteme. Verstehen grundsätzlicher Herausforderungen der Robotik und überblicksmäßige Kenntnisse verschiedener methodischer Ansätze. |
| Inhalt: | Die Vorlesung behandelt die Grundlagen autonomer mobiler Roboter. Zu den Themen gehören Hardware-Komponenten, Sensorik und Aktorik, Informationsverarbeitung, Software-Architekturen, Weltmodellierung Verhaltenssteuerung, etc. Die Lehrveranstaltung besteht aus wöchentlichen Vorlesungen und Übungen. |
| Studien-/Prüfungsleistungen: | Projektarbeit/mdl. Prüfung/schriftl. Prüfung |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Avatare in Serious Games |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor |
| Lehrform/SWS: | 4SWS: 2SWS Vorlesung & 2SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | 3D Modellierung |
| Angestrebte Lernergebnisse | Der/die Studierende kann: <ul style="list-style-type: none"> - Techniken der Animationen von Avataren in Serious Games beschreiben - sich durch Nutzung von relevanten Werkzeugen (Software) grundlegende Kenntnisse in der Erstellung von Avataren und deren Animationen erarbeiten. - eine einfache erste Simulation einer Avatar-Mensch-Interaktion konfigurieren |
| Inhalt: | <ul style="list-style-type: none"> - Avatar Modeling in Mekehuman - Unity and Blender introduction - Files export and import - UV Maps introduction - Texturing - Rigging and skinning - Learn the basic mecanim system in Unity - Motionbuilder introduction - Motion capturing using mocap lab - Computer human interactions using Kinect |
| Studien-/Prüfungsleistungen: | Studienleistung: Werden vorausgesetzt, siehe moodle Prüfungsleistung: Hausarbeit (30% semesterbegleitende Projekte und 70% Endprojekt) Weitere Informationen siehe moodle |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Betriebliche Anwendung von Internettechnologien |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Jan Marco Leimeister |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor und andere |
| Lehrform/SWS: | 4 SWS Vorlesung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Informationswissenschaften I |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> - Kenntnisse im Systementwicklungsprozess - Fähigkeit, Anforderungen an Software-Systeme strukturiert zu erheben - Kenntnisse in den Bereich Content-Management, Wissensmanagement und Computer-Supported-Cooperative Work bzw. Collaboration Engineering - Kenntnisse in der rechnergestützten Aus- und Weiterbildung / elearning / blended learning - Fähigkeit, Mensch-Computer-Interaktion zu bewerten und zu gestalten - Fähigkeit, den wirtschaftlichen Nutzen von Software im betriebswirtschaftlichen Anwendungszusammenhang zu analysieren. |
| Inhalt: | <ul style="list-style-type: none"> - Systementwicklungsprozess - Anforderungserhebung für Software-Systeme - Content Management - Wissensmanagement - Computer-Supported-Cooperative Work bzw. Collaboration Engineering - elearning / blended learning - Interaktionsdesign - Kosten- Nutzen- Analyse |
| Studien-/Prüfungsleistungen: | Klausur (2 Std.) oder Hausarbeit (20 S.) oder Referat (20 Min.) mit schriftl. Ausarbeitung (ca. 12 S.) oder mündliche Prüfung (30 Minuten) |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | C++ für Fortgeschrittene |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik, Anwendungsgebiet Software Tools |
| ggf. Kürzel | C++ FG |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik |
| Lehrform/SWS: | 2 SWS Vorlesung, 2SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundkenntnisse einer objektorientierten Programmiersprache |
| Angestrebte Lernergebnisse | Gründliche Kenntnisse der erweiterten Sprachelemente in C++, Verständnis für optimierte Programmausführung, Vertiefen der Programmierkonzepte, gute Fertigkeiten bei der Entwicklung komplexerer Programme bis etwa 600 Zeilen, Fertigkeiten im Umgang mit Werkzeugen für Programmiersprachen, überblicksmäßige Kenntnisse von größeren Softwarepaketen. Entwicklung der Fähigkeit zur selbstständigen Problemlösung und Projektorganisation |
| Inhalt: | Umgang mit Entwicklungsumgebungen, größeres Softwarepaket, Werkzeuge zur Programmierung, Erweiterte Konzepte der Programmiersprache C++ |
| Studien-/Prüfungsleistungen: | Studienleistungen werden vorausgesetzt, siehe moodle Prüfungsleistung: Klausur |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Compilerbau und Reverse Engineering |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Anwendungsgebiet Software Tools |
| ggf. Kürzel | SE |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Albert Zündorf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatiker |
| Lehrform/SWS: | Vorlesung, 4 SWS, 15 Teilnehmer |
| Arbeitsaufwand: | Stunden Präsenz Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen Informatik |
| Angestrebte Lernergebnisse | Lernziele: Kontextfreie Grammatiken, Parsertabellen, Scanner- und Parsergeneratoren, synthetische und inherite Attribute, Symboltabellen, Byte Code Generierung, Cross Referencing, Pattern Erkennung, Laufzeitanalysen, Refactorings / Fachkompetenz 60%, Methodenkompetenz 30%, Systemkompetenz 10%. |
| Inhalt: | Die Vorlesung behandelt zunächst den klassischen Compilerbau, also das Erstellen von Grammatiken, die Verwendung von Parsergeneratoren, die Ableitung von Symboltabelleninformationen und die Generierung von Byte Code. Darauf aufbauend werden wir uns mit Anwendungen des Compilerbaus für das Reverse Engineering beschäftigen, also mit Pattern Erkennung, Code Instrumentierung und Refactoring Operationen <ul style="list-style-type: none"> • Kontextfreie Grammatiken • Parsertabellen • Scanner- und Parsergeneratoren • Synthetische und inherite Attribute • Symboltabellen • Byte Code Generierung • Cross Referencing • Pattern Erkennung • Laufzeitanalysen • Refactorings |
| Studien-/Prüfungsleistungen: | Bearbeitung von Übungsaufgaben |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Computational Intelligence in der Automatisierung |
| ggf. Modulniveau | Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Robotik |
| ggf. Kürzel | CIA |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Andreas Kroll (FB Maschinenbau) |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Mechatronik (Bachelor), Mechatronik (Master), Maschinenbau (Bachelor), Maschinenbau (Master), |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz, 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Kenntnisse Regelungstechnik |
| Angestrebte Lernergebnisse | <p>Die Studierenden verstehen die grundlegenden, Begriffe, Konzepte und Methoden der Computational Intelligence (CI) mit ihren drei Teilgebieten Fuzzy-Logik, künstliche Neuronale Netze und Evolutionäre Algorithmen.</p> <p>Die Studierenden werden befähigt, einfache CI-Anwendungen selbstständig und systematisch zu erstellen. Des Weiteren erwerben Studierende eine ausreichende Kompetenz, um die Eignung von CI-Methoden zur Lösung einer technischen Aufgabe abschätzen zu können. Sie können die entsprechende technisch-wissenschaftliche Literatur lesen.</p> |
| Inhalt: | <ul style="list-style-type: none"> • Was bedeutet Computational Intelligence und was ist das besondere an ihr? • Fuzzy-Logik und Fuzzy-Systeme <ul style="list-style-type: none"> - Mustererkennung - Modellbildung, Regelung - Anwendungsbeispiele • Künstliche Neuronale Netze <ul style="list-style-type: none"> - Allgemeine Prinzipien - Netzwerke vom MLP-, RBF- und SOM-Typ - Anwendungsbeispiele • Evolutionäre Algorithmen <ul style="list-style-type: none"> - Allgemeine Prinzipien - Genetische Algorithmen, Evolutionäre Strategien - Genetisches Programmieren - Anwendungsbeispiele • Hybride Systeme • Realisierungsaspekte und Tools • Ausblick: Schwarmintelligenz & künstliche Immun-systeme |
| Studien-/Prüfungsleistungen: | Klausur (120 min.) oder mündliche Prüfung (30 min.) |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Computergraphik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | CG |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik |
| Lehrform/SWS: | Vorlesung, 2 SWS, Übung, 2 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Kenntnisse in einer objektorientierten Programmiersprache |
| Angestrebte Lernergebnisse | Erwerb konzeptueller Kenntnisse interaktiver 3D- Computergraphik durch das Erlernen mathematischer und algorithmischer Konzepte von 3D Graphikanwendungen. Fertigkeiten in der Graphikprogrammierung durch praktische Programmierung mit OpenGL. Erlernen der Planung und anschließenden Erstellung von eigenen Programmen, realisiert mittels OpenGL. Grundlegende Kenntnisse im Bereich Visualisierung und Simulation durch Vermittlung der Zusammenhänge von Computergraphik-Grundlagen und deren weiterführender Nutzung am Beispiel einer Game Engine. |
| Inhalt: | Einführung in OpenGL Theoretische Grundlagen der Computergraphik Einsatz objektorientierter Ansätze in der Graphik- Programmierung Konzeptvisualisierung mit Game Engines |
| Studien-/Prüfungsleistungen: | Studienleistung: Regelmäßige Bearbeitung v. Übungsaufgaben Prüfungsleistung: Klausur |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Data Mining für Technische Anwendungen |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Basis Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Knowledge & Data Engineering |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker |
| Angestrebte Lernergebnisse | Kenntnisse: Aufgaben und Schritte des Data Mining, wesentliche Paradigmen aus dem Bereich des Data Mining Fertigkeiten: praktischer Einsatz der Paradigmen (geübt unter Verwendung von Matlab oder RapidMiner) Kompetenzen: Bewertung von praktischen Anwendungen der Paradigmen, selbständige Entwicklung von einfachen Anwendungen |
| Inhalt: | Die Vorlesung beschäftigt sich hauptsächlich mit Algorithmen des Data Mining wie sie in technischen Anwendungen benötigt werden. Der Schwerpunkt liegt auf Klassifikationstechniken. Folgende Themen werden besprochen: Grundlagen und Datenvorverarbeitung, Merkmalsselektion, lineare Klassifikatoren (u.a. Perzeptron-Lernen, lineares Ausgleichsproblem, Fisher-Kriterium), nichtlineare Klassifikatoren (u.a. Support Vector Machines, RBF-Netze, Generative Klassifikatoren, Relevance Vector Machines), Bayessche Netze, Ensembletechniken, Grundlagen des Spatial Data Mining und des Temporal Data |
| Studien-/Prüfungsleistungen: | Schriftliche Prüfung (120 min) oder mündliche Prüfung (ca. 20 min.) |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Datenschutzrecht |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Alexander Roßnagel |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 2 SWS Vorlesung oder 2 SWS Seminar |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden Selbststudium: 60 Stunden |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Angestrebte Lernergebnisse | Studierende haben die wichtigsten geltenden Vorschriften und ihr systematisches Zusammenspiel kennen gelernt. Sie kennen die technischen, politischen und wirtschaftlichen Grundlagen des elektronischen Rechtsverkehrs. Sie sind in der Lage, praktische Fälle mit einschlägigen Rechtsproblemen zu lösen. |
| Inhalt: | Für den E-Commerce relevantes Datenschutzrecht, verfassungsrechtliche Grundlagen, einschlägige Datenschutzgesetze und Abgrenzung der Anwendungsbereiche, Zulässigkeit verschiedener Formen der Verarbeitung personenbezogener Daten, Datenschutzprinzipien der Transparenz, der Zweckbindung, der Erforderlichkeit, der Datensparsamkeit, der Datensicherheit und der Mitwirkung der Betroffenen |
| Studien-/Prüfungsleistungen: | Klausur oder Referat |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Design Patterns |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Anwendungsgebiet Software Tools |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Albert Zündorf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatiker |
| Lehrform/SWS: | Vorlesung, 4 SWS, 20 Teilnehmer |
| Arbeitsaufwand: | 180h: 60h Präsenzzeit 120h Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen Informatik |
| Angestrebte Lernergebnisse | Die Studierenden können Architekturen mit Hilfe von Design Patterns diskutieren und entwerfen. Die Studierenden können Design Patterns in neuen Programmen verwenden und richtig implementieren. Die Studierenden erkennen Designprobleme und können beurteilen, welche Design Patterns zur Verbesserung verwendet werden könnten. |
| Inhalt: | <p>Software-Entwurf ist eine anspruchsvolle Tätigkeit und erfordert Erfahrungen. Qualitativ-hochwertige und wiederverwendbare Software zu erstellen ist schwer. Die Idee von "Design Patterns" (dt. Entwurfsmuster) ist es, Erfahrungen von Experten zu sammeln und so darzustellen, dass diese leicht auf neue Aufgaben übertragen werden können.</p> <p>Die Idee stammt ursprünglich aus der Architektur und geht vor allem auf den Architekten Christopher Alexander zurück.</p> <p>Er umschreibt Design Patterns wie folgt: "Jedes Muster beschreibt ein Problem, das in unserer Arbeitsumgebung immer und immer wieder auftaucht, und dann beschreibt es den Kern einer Lösung dieses Problems so, dass man diese Lösung tausendfach einsetzen kann, ohne das Problem zweimal in identischer Weise gelöst zu haben."</p> <p>Die Entwurfsmuster von Alexander präsentieren Lösungen für den Entwurf von Häusern und Städten. Entwurfsmuster im Software-Engineering zeigen bewährte Lösungen für die Konstruktion von Software. Inhalt dieser Vorlesung sind Grundlagen und eine Übersicht der verschiedenen Entwurfsmuster-Ansätze. Eine Reihe von Entwurfsmustern für die Softwarekonstruktion werden vorgestellt und es wird dargelegt, wie die jeweiligen Muster einzeln und vor allem als "Mustersprache" helfen, Software flexibler und vor allem wiederverwendbar zu erstellen.</p> |
| Studien-/Prüfungsleistungen: | Benotete Hausaufgaben |

| | |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Digitale Signalverarbeitung mit integrierten Schaltungen |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Kürzel | |
| ggf. Untertitel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Sommersemester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Elektrotechnik (Bachelor) Informatik (Bachelor) |
| Lehrform/SWS: | 4 SWS: 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit, 120 h Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik; zusätzlich wünschenswert: VHDL-Kurs oder äquivalente LV, Signalverarbeitung mit Mikroprozessoren. |
| Angestrebte Lernergebnisse | Die/der Lernende kann – wichtige Komponenten und Algorithmen der digitalen Signalverarbeitung (DSV) nennen und erläutern, – Architekturen für Algorithmen der DSV entwerfen, – Implementierung und Test von Architekturen und Algorithmen der DSV durchführen. |
| Inhalt: | Grundlagen der digitalen Signalverarbeitung, Überblick über Aufbau und Funktion von VLSI-Schaltungen und FPGAs, Zahlendarstellungen, Realisierung arithmetischer Schaltungen, Implementierungskonzepte datenpfadorientierter Algorithmen, Optimierungsverfahren bezüglich Fläche, Geschwindigkeit und Verlustleistung, Realisierung ausgewählter Komponenten (Digitale Filter, FFT). |
| Klausur, Studienleistungen (b/bn): Übungsaufgaben | Mündl. Prüfung (etwa 40 Min.) |
| Medienformen: | Folien/Beamer, Tafel, Rechnerübung |
| Literatur: | – Oppenheim/Schafer; Zeitdiskrete Signalverarbeitung; 2. Auflage (2004) – Kammeyer; Digitale Signalverarbeitung; 7. Auflage (2009) – Parhi: VLSI Digital Signal Processing Systems – U. Meyer-Baese: Digital Signal Processing for Field Programmable Gate Arrays Weitere Literatur wird in der Vorlesung bzw. auf der Homepage des Fachgebiets bekannt gegeben. |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Digitale Systeme |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Prozessor- und Rechner-technologie |
| ggf. Untertitel | Digitaltechnik II |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | nach Absprache |
| Zuordnung zum Curriculum | Elektrotechnik (BSc.); Informatik (BSc.); Mechatronik (BSc.) |
| Lehrform/SWS: | 4 SWS: 3 SWS Vorlesung; 1 SWS Übung |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit; 120 h Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik (Digitaltechnik I) |
| Angestrebte Lernergebnisse | Verständnis spezieller Aspekte des Entwurfs digitaler Schaltungen. Studenten sollen in die Lage versetzt werden, komplexe digitale Schaltungen zu planen, zu optimieren und zu analysieren. |
| Inhalt: | Logiksynthese, Zustandsautomaten, Synchronisation, Pipelinestrukturen, Computerarithmetik. |
| Klausur, Studienleistungen (b/bn): | Klausur oder mündl. Prüfung oder Hausarbeit |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Echtzeitsysteme |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Prozessor- und Rechnertechnologie |
| ggf. Lehrveranstaltungen | Ersetzt die frühere Veranstaltung „Prozessinformatik“. Es kann nur eine der beiden Veranstaltungen ins Studium eingebracht werden. |
| Studiensemester: | Ab 6. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Rechnerarchitektur, Betriebssysteme |
| Angestrebte Lernergebnisse | Kenntnisse: wichtigste Grundlagen der Echtzeitverarbeitung, speziell Hardware und Echtzeitbetriebssysteme Fertigkeiten: Programmierung einer Echtzeitanwendung Kompetenzen: Bewertung von praktischen Anwendungen, Konzeption einfacher Echtzeitsysteme |
| Inhalt: | Grundlagen von Echtzeitsystemen, Hardwareanforderungen (u.a. Interrupttechnik, Timer), Echtzeitbetriebssysteme (u.a. Schedulingtechniken wie Rate Monotonic Scheduling oder Earliest Deadline First, Prioritätsinversion, Prioritätsanhebung), Softwareanforderungen & Programmiersprachen, Entwurfsmethodik (u.a. Endliche Automaten, Petri-Netze), Performanzbewertung |
| Studien-/Prüfungsleistungen: | Schriftliche Prüfung (180 min.) oder mündliche Prüfung (ca. 20 min.) |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Aktorik |
| ggf. Modulniveau | Anwendungsgebiet Robotik |
| ggf. Kürzel | EAK |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Dr. Hanns Sommer (FB Maschinenbau) |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mechatronik (Bachelor), Maschinenbau (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 45 Stunden Präsenz 75 Stunden Selbststudium, |
| Kreditpunkte: | 4 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | Den Studierenden wird gezeigt wie, ausgehend von Grundprinzipien der Physik, eine Erzeugung von Wirkungen in mechatronischen Systemen möglich ist. Es wird besonderer Wert daraufgelegt, den Studierenden eine anschauliche Vorstellung der physikalischen Wirkungsprinzipien zu vermitteln. Sie sollen dadurch die Fähigkeit erlangen, die Ideen von Aktorkonzepten zu verstehen, um selbst solche Konzepte entwickeln zu können. |
| Inhalt: | Stellung eines Aktors im mechatronischen System; Anforderungen an einen Aktor; Prinzipieller Aufbau eines Aktors; Elektromagnetische Aktoren; Fluidtechnische Aktoren; Unkonventionelle Aktoren; (Thermobimetalle, Memory-Legierungen, Dehnstoff-Elemente, Piezo-Aktoren etc.); Elektronische Aktoren; Mikroaktoren; Biophysikalische Aktoren; Smart Structures, Aktorfelder. |
| Studien-/Prüfungsleistungen: | schriftliche Prüfung |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die formale Verifikation |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt, Anwendungsgebiet Software Tools |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Martin Lange |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor |
| Lehrform/SWS: | Vorlesung/Übung: 4 SWS |
| Arbeitsaufwand: | 180 h: 60 Stunden Präsenzzeit 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | |
| Empfohlene Voraussetzungen: | Theoretische Informatik – Logik, Theoretische Informatik – Berechenbarkeit und Formale Sprachen |
| Angestrebte Lernergebnisse | Die Veranstaltung vermittelt die Notwendigkeit des Einsatzes formaler Methoden in der Entwicklung korrekter und sicherer Software anhand einer ausgesuchten Verifikationsmethode, des Model Checkings. Nach erfolgreichem Abschluss sollen die Teilnehmer grundlegende Methoden zum Nachweis der Korrektheit von Programmen kennengelernt haben und selbst einsetzen können. Die Veranstaltung bereitet auf eine Tätigkeit in der Software-Entwicklung, insbesondere auf den Einsatz formaler Methoden darin, vor. |
| Inhalt: | <ul style="list-style-type: none"> – Formale Verifikation – Prozessalgebra – Temporale Logik – Model Checking – Büchi-Automaten – Binary Decision Diagrams – Verifikationstools |
| Studien-/Prüfungsleistungen: | Studienleistung: Bearbeitung der wöchentlichen Übungsaufgaben und aktive Teilnahme in den Übungen. Prüfungsleistung: Mündliche Prüfung (ca. 30 Min.) / Klausur (ca. 120 Min.) je nach Teilnehmerzahl |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Programmierung mit C++ |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik |
| ggf. Kürzel | C++ |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Vorlesung 2 SWS, Übung 2 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen | keine |
| Angestrebte Lernergebnisse | Gründliche Kenntnisse der Sprachelemente in C++, Verständnis für Abläufe im Rechner bei Programmausführung, Verstehen grundlegender Programmierkonzepte (z.B. Objektorientierung), gute Fertigkeiten bei Entwicklung prozeduraler Programme bis etwa 200 Zeilen, Fertigkeiten in objektorientierter Programmierung, überblicksmäßige Kenntnisse der Grundkonzepte der Software-Entwicklung und Umgang mit Entwicklungsumgebungen. Kenntnis von Anwendungen mit C++, Entwicklung von Fähigkeit zur selbstständigen Problemlösung und Projektorganisation |
| Inhalt: | Grundlagen in der Programmiersprache C++: Variablen, Pointer, Datentypen, Operatoren, Funktionen, Kontrollstrukturen, Klassen, Vererbung, Bibliotheken |
| Studien-/Prüfungsleistungen: | Studienleistungen werden vorausgesetzt, siehe moodle Prüfungsleistung: Form: Klausur Dauer: ca. 120 Min |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Umweltinformatik |
| ggf. Modulniveau | Basis Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortlicher: | Prof. Dr. Joseph Alcamo PD Dr. Rüdiger Schaldach |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2 SWS Vorlesung |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | Verständnis der grundlegenden Methoden der Umweltinformatik. |
| Inhalt: | Übersicht über ICT Methoden, um den Schutz und die nachhaltige Bewirtschaftung von Umweltsystemen zu unterstützen; Umweltinformationssysteme und Datenbanken, Geographische Informationssysteme (GIS) sowie Methoden der Modellbildung und Simulation. |
| Studien-/Prüfungsleistungen: | Klausur (60 Min) |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in die Umweltwissenschaften |
| ggf. Modulniveau | Basis Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortlicher: | Prof. Dr. Joseph Alcamo PD Dr. Rüdiger Schaldach Dr. Martina Flörke |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2 SWS Vorlesung |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | Kenntnisse über die Funktionsweise von Umweltsystemen und deren Beeinflussung durch den Menschen. |
| Inhalt: | Funktionsweise von Prozessen in den Bereichen Klima, Hydrologie und Ökologie; Mensch-Umwelt-Beziehungen; Maßnahmen zur Minderung von Umweltbelastungen. |
| Studien-/Prüfungsleistungen: | Klausur (60 Min) |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Einführung in Matlab/Simulink |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik Basis Anwendungsgebiet Modellierung und Simulation |
| ggf. Kürzel | EMS |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Arno Linnemann |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 1 SWS Vorlesung, 0,5 SWS Übung |
| Arbeitsaufwand: | 60 Stunden, davon 22 Stunden Präsenz |
| Kreditpunkte: | 2 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Kenntnis einer Programmiersprache; Kenntnisse entsprechend der Lehrinhalte des Moduls „Lineare Algebra“; Hilfreich sind ferner Kenntnisse entsprechend der Lehrinhalte des Moduls „Technische Systeme im Zustandsraum“ |
| Angestrebte Lernergebnisse | Der/die Lernende kann die Syntax grundlegender Funktionen und Strukturen angeben, die Funktionsweise von vorhandenen Matlab-Programmen und Simulink-Modellen erfassen, interpretieren und modifizieren, eigene Programme und Modelle entwickeln, die Software-Dokumentation zur Erweiterung der eigenen Kenntnisse nutzen. |
| Inhalt: | Im Rahmen einer Einführung in die Software "Matlab" und ihre Ergänzungen "Control System Toolbox" sowie "Simulink" werden die folgenden Themen behandelt: <ul style="list-style-type: none"> - Grundbegriffe - Matrizenrechnung - Datenstrukturen, Grafik - Logische Verknüpfungen - Funktionen, Optimierung - Analyse linearer Systeme - Simulation nichtlinearer Systeme |
| Studien-/Prüfungsleistungen: | Prüfungsleistung: Klausur (45 min.) oder mündl. Prüfung (30 min.) Studienleistung: Übungsaufgaben, Hausarbeit |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Elektrische und elektronische Systeme im Automobil |
| ggf. Modulniveau | Anwendungsgebiet Modellierung und Simulation |
| ggf. Kürzel | EES |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. Ludwig Brabetz |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Elektrotechnik (Bachelor) Informatik (Bachelor) Mechatronik (Bachelor) Wirtschaftsingenieurwesen Elektrotechnik (Bachelor) |
| Lehrform / SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen Fahrzeugtechnik, Elektrotechnik, Informatik, Nachrichtentechnik, Regelungstechnik |
| Angestrebte Lernergebnisse: | Die Studierenden können <ul style="list-style-type: none"> - die Funktion und den Entwicklungsprozesses von automotiven Systemen erläutern, - die Vernetzung von Systemen beschreiben, - technische Synergien aufzeigen, - Risiken und wirtschaftliche Zusammenhänge erfassen, - den Bezug bereits erlernter Basiskompetenzen zu Anwendungen und deren technischen Umsetzungen und Randbedingungen herstellen. |
| Inhalt: | Teil 1: Produktentstehungsprozess des Automobils, Projektmanagement, Fahrzeugbordnetze und elektrische Energieversorgung im Auto, Fahrzeugelektrik: Aktuatorik, Licht, Absicherung, Schalten, Grundlagen Bussysteme, Elektronische Systeme 1: Antriebsstrang, Alternative Antriebe Teil 2: Fahrzeugdynamik (ABS/ESP/Lenkung/Dämpfung), Insassenschutz, Security, MMI, Assistenzsysteme, Bussysteme 2, Diagnoseverfahren und Risiko-Assessment, Werkzeuge für die Entwicklung mechatronischer Systeme: CASE/CAX, Validierung, Architektur, Zukunftstrends |
| Studien-/Prüfungsleistungen: | Teil 1: Klausur, 100 min, Teil 2: Klausur, 100 min |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Entwurf und Analyse von Algorithmen |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Kürzel | |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 5. |
| Modulverantwortlicher: | Prof. Dr. Friedrich Otto |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Mathematik (Master), Informatik (Bachelor), Informatik (Master) |
| Lehrform / SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | Allgemein: Die Veranstaltung behandelt Strategien zum Entwurf und zur Analyse von Algorithmen. Berufsvorbereitung: Die Veranstaltung bereitet auf den Einsatz in der Softwareentwicklung vor. |
| Inhalt: | <ul style="list-style-type: none"> - Rechenzeit- und Speicherplatzbedarf von Algorithmen, - Strategien zum Entwurf und zur Analyse von Algorithmen, - Methoden zur Herleitung unterer Schranken, - Approximations-Algorithmen, probabilistische Algorithmen, parallele Algorithmen |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung (ca. 30 Min.) nach erfolgreicher Teilnahme an den Übungen |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Ereignisdiskrete Systeme und Steuerungstheorie |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, wenn das Anwendungsgebiet Modellierung und Simulation gewählt wurde. Anwendungsgebiet Modellierung und Simulation |
| ggf. Kürzel | ESS |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Olaf Stursberg |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | Der/die Lernende kann: <ul style="list-style-type: none"> - schrittweise ablaufende Prozesse durch ereignis-diskrete Modelle beschreiben, - ereignisdiskretes dynamisches Verhalten definieren, - Eigenschaften ereignisdiskreter Systeme analysieren, - Steuerungen auf der Basis von Automaten und Petri-Netzen entwerfen berechnen, - nichtdeterministische und stochastische Prozesse durch Markov-Ketten beschreiben, - Algorithmen zum Steuerungsentwurf interpretieren, und Steuerungsprogramme in Form genormter Sprachen darstellen. |
| Inhalt: | <ul style="list-style-type: none"> - Einführung in ereignisdiskretes Systemverhalten - Modellierung mit endlichen Automaten, - Steuerungssynthese mit endlichen Automaten - Definition, Analyse und Steuerungssynthese mit Petri-Netzen - Hierarchischer Systementwurf mit Statecharts - Stochastische ereignisdiskrete Modelle - Echtzeitmodelle - Simulation ereignisdiskreter Systeme - Stabilität gesteuerter Systeme und Systemanalyse durch Model-Checking - Steuerungssprachen für SPS |
| Studien-/Prüfungsleistungen: | Klausur (90 min.) oder mündliche Prüfung (30 min.) |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Formale Sprachen und Automaten I |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Friedrich Otto |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Informatik (Master), Mathematik (Diplom) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | EinfProg, Diskrete Strukturen |
| Angestrebte Lernergebnisse | Allgemein: Die Veranstaltung behandelt fortgeschrittene Techniken aus dem Gebiet der Automatentheorie und der Formalen Sprachen. – Kompetenzen: Das vermittelte Methodenwissen hilft den Studierenden, Automatenmodelle und Grammatiktypen zur Beschreibung und Analyse von formalen Sprachen auszuwählen und einzusetzen. – Berufsvorbereitung: Die Veranstaltung bereitet auf den Einsatz in der Forschung und bei der Softwareentwicklung vor. |
| Inhalt: | Voraussichtlich werden folgende Themen behandelt: – kontext-freie Sprachen: Gleichungssysteme, Normalformen, Entscheidungsprobleme, – wachsend-kontext sensitive Sprachen, – Church-Rosser Sprachen – Zweikeller-Automaten – Grammatiken mit kontrollierten Ableitungen |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung (ca. 30 Min.) nach erfolgreicher Teilnahme an den Übungen |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Funktionale Programmierung |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Anwendungsgebiet Software Tools |
| ggf. Kürzel | FP |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Claudia Fohry |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor u. andere |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | EinfProg, Diskrete Strukturen |
| Angestrebte Lernergebnisse | Gründliche Kenntnisse einer funktionalen Programmiersprache (z.B. Haskell), überblicksmäßige Kenntnisse weiterer funktionaler Programmiersprachen, Verstehen von Konzepten der funktionalen Programmierung auch im Vergleich zur imperativen Programmierung, gute Fertigkeiten in Entwicklung funktionaler Programme bis ca. 200 Zeilen, Kenntnis von Anwendungen der funktionalen Programmierung, Entwicklung von Fähigkeit zur Teamarbeit und Projektorganisation |
| Inhalt: | Funktionale Programmierung am Beispiel der Sprache Haskell einschließlich fortgeschrittener Konzepte (z. B. Monaden). Weitere funktionale Sprachen werden im Überblick vorgestellt. Schwerpunkt: Bewertung unterschiedlicher Programmierkonzepte, auch im Vergleich zur imperativen Programmierung |
| Studien-/Prüfungsleistungen: | Projektarbeit oder mündliche Prüfung (20 – 40 min.) oder schriftliche Prüfung |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Geographische Informationssysteme (GIS) |
| ggf. Modulniveau | Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. Joseph Alcamo Dr. Martina Flörke |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2SWS Vorlesung/Übung |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen der Umweltinformatik |
| Angestrebte Lernergebnisse: | Die Studierenden erlernen die theoretischen Grundlagen Geographischer Informationssysteme (GIS) sowie das konzeptionelle Design von GIS-Projekten. In der Übung werden an Hand konkreter umweltwissenschaftlicher Fragestellungen die theoretischen Kenntnisse der Vorlesung praktisch umgesetzt. |
| Inhalt: | Geographische Informationssysteme, Geodatenverwaltung, Geodatenanalyse, Interpretation räumlicher Daten, Fallstudien |
| Studien-/Prüfungsleistungen: | Klausur, Projektarbeit |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Graphische Simulation |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik, |
| ggf. Kürzel | GS |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | Vorlesung, 2 SWS, Übung 2 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundkenntnisse einer objektorientierten Programmiersprache |
| Angestrebte Lernergebnisse | <p>Umfassende konzeptuelle Kenntnisse der Programmierung von graphischen Simulationen, speziell aus dem Bereich Serious Games.</p> <p>Ausgeprägte Fertigkeiten im praktischen Umgang mit Software zur Erstellung eines Serious Games, beispielsweise per Game Engine. Somit praktische Umsetzung der erworbenen konzeptuellen Kenntnisse.</p> <p>Grundlegende Fähigkeiten zur Planung, Erstellung und Nutzung von benötigten Requisiten (Assets) für graphische Simulationen.</p> <p>Breit gefächerte Kenntnisse und Fertigkeiten in der Anwendung von Komponenten graphischer Simulationen, wie beispielsweise Assets, Animationen, Sound, Physik und anderen.</p> <p>Entwicklung von Fähigkeiten zur selbständigen Problemlösung und Projektorganisation.</p> <p>Entwicklung von Teamfähigkeit durch die Organisation, gemeinsame Bearbeitung und Einteilung von Aufgabenstellungen.</p> |
| Inhalt: | <ul style="list-style-type: none"> • Erlernen der grundlegenden Programmier-techniken und Konzepte graphischer Echtzeitsimulation • Szenegraphensysteme • Erstellung eigener Anwendungen aus dem Bereich Game-Engines und Serious Gaming |
| Studien-/Prüfungsleistungen: | <p>Studienleistung: Regelmäßige Bearbeitung v. Übungsaufgaben</p> <p>Prüfungsleistung: Hausarbeit</p> |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Grundlagen der Algebra und Computeralgebra |
| ggf. Modulniveau | Basis Anwendungsgebiet Computational Mathematics |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Dozenten der Mathematik |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mathematik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, Lineare Algebra |
| Angestrebte Lernergebnisse | Die Studierenden erhalten exemplarisch Einblick in ein konkretes Anwendungsgebiet innerhalb der Mathematik und vertiefen zugleich ihre Kenntnisse und Fertigkeiten. Die mathematischen Fächer fördern unter anderem abstraktes Denken, Fertigkeit im Umgang mit Formalismen sowie Problemlösungskompetenz. Die Vorlesung soll die Studenten in die Lage versetzen, eigenständig Programme der Computeralgebra zu entwerfen und existierende Programme beurteilen zu können. Die Veranstaltung bereitet die Studenten auch auf eine mögliche Bachelor- oder Masterarbeit im Bereich der Computeralgebra vor. |
| Inhalt: | Folgende Themen werden behandelt: <ul style="list-style-type: none"> • Programmieren in Computeralgebrasystemen • Zahlssysteme und Ganzzahlarithmetik • Modulare Arithmetik: Rechnen in Restklassenringen • Codierungstheorie und Kryptographie • Polynomarithmetik: Rechnen mit Polynomen und |
| Studien-/Prüfungsleistungen: | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistung: Schriftliche (120 – 180 min.) oder mündliche (30 – 40 min.) Prüfung |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Grundlagen der angewandten Kryptologie |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | GdaK |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortlicher: | Dr. Arno Wacker |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Diskrete Strukturen, Einführung in die Programmierung für Informatik |
| Angestrebte Lernergebnisse: | Fundierte Kenntnisse über die grundlegende Funktionsweise von verschiedenen Algorithmen zur Nachrichtenverschlüsselung (Verständnis von Substitution/Transposition aber auch der mathematischen Grundlagen der modernen asymmetrischen Verfahren). Verständnis der verschiedenen Facetten des Begriffs Sicherheit: Ausgehend von den Verfahren zur Verschlüsselung, der Schlüsselgenerierung und digitaler Signaturen werden auch die Begriffe der Hashbildung, Authentifizierung und Zero-Knowledge erlernt. Fertigkeit um die Sicherheit von verschiedenen Verfahren selbst zu analysieren und einzuschätzen. |
| Inhalt: | Es werden verschiedene Methoden zur Verschlüsselung von Nachrichten vorgestellt (Kryptographie). Es wird auf die unterschiedlichen Verfahren die im Laufe der Zeit erfunden und verwendet wurden eingegangen. Dies beinhaltet klassische Verfahren (z.B. Caesar, Vigenere, Playfair), mechanische Verfahren (Enigma) und moderne symmetrische (DES, AES, RC4) und asymmetrische Verfahren (DH, RSA, ElGamal). Dabei wird parallel auch immer auf die Sicherheit bzw. die Angriffsmöglichkeiten der Verfahren eingegangen (Kryptoanalyse). Schwerpunkt: Bewertung der Sicherheit von den verschiedenen Verfahren zur Nachrichtenverschlüsselung, Steigerung des Sicherheitsbewusstseins |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Grundlagen der Elektrotechnik II für Informatiker |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Kürzel | |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortlicher: | Prof. Dr. Witzigmann |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) |
| Lehrform / SWS: | Vorlesung 2 SWS Übung 1 SWS Tutorium 2 SWS Praktikum 1 SWS |
| Arbeitsaufwand: | Vorlesung und Übung: Präsenzzeit: 3 SWS Selbststudium: 2 SWS Praktikum: Präsenzzeit: 1 SWS Selbststudium: 1 SWS |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Gute Kenntnisse der Mathematik der Oberstufe, Analysis, Algebra |
| Angestrebte Lernergebnisse: | <p>Kenntnisse: Zusammenhänge der komplexen Wechselstromlehre – Bauelemente, Signalformen, einfache Grundschaltungen. Strukturieren von elektrotechnischen und energietechnischen Problemen.</p> <p>Fähigkeiten: Verwendung von Transformationstechniken um lineare passive Schaltungen breitbandig mathematisch zu beschreiben und zu berechnen.</p> <p>Elektrotechnisches Praktikum: Die Studierenden können die Grundlagen der Elektrotechnik anwenden, einfache elektrotechnische Grundschaltungen aufbauen, messtechnische Geräte bedienen, elektrotechnische Größen messtechnisch erfassen und durchgeführte Messungen interpretieren und dokumentieren.</p> |

| | |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Inhalt:</p> | <p>Wechselstromlehre: Zeitabhängige Ströme und Spannungen, Eingeschwungene Sinusströme und -spannungen in linearen RLC-Netzen, Resonanz in RLC-Schaltungen, Einfache Filterschaltungen, Resonanzkreise, Leistung und Energie in Wechselstromkreisen</p> <p>Fourier Reihen & Fourier Transformation: die Fourier-Reihe, Mehrfrequente Vorgänge in linearen Netzwerken, Harmonische Analyse, Fourier-Transformation, Fourier-Integral, Abtastung im Zeitbereich, Diskrete Fourier-Transformation Schaltvorgänge</p> <p>Elektrotechnisches Praktikum: 3 Grundlagenversuche zur Einführung in das Messen mit Multimeter und Oszilloskop aus dem Themenbereich: Strom/Spannungskennlinie, Widerstand, Diode und dielektrische und magnetische Werkstoffe.</p> |
| <p>Studien-/Prüfungsleistungen:</p> | <p>Regelmäßiges Bearbeiten von Übungs- und Tutoriumsaufgaben. Klausur</p> |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Grundlagen der Regelungstechnik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, wenn das Anwendungsgebiet Modellierung und Simulation gewählt wurde. Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Modellierung und Simulation, Anwendungsgebiet Robotik |
| ggf. Kürzel | GRT |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Olaf Stursberg |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor) |
| Lehrform/SWS: | Vorlesung, 3 SWS, Übung 1 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker |
| Angestrebte Lernergebnisse | Der/die Lernende kann: <ul style="list-style-type: none"> - Grundlegende Eigenschaften dynamischer Systeme erläutern und einordnen, - Dynamisches Verhalten durch Übertragungsfunktionen darstellen, - Ziele der Regelung technischer Prozesse formulieren, - Methoden des Reglerentwurfes für skalare, lineare zeitinvariante Systeme nutzen, - die Eignung bestimmter Reglertypen für gegebene Systeme und Anforderungen bewerten und erhaltene Regelungsergebnisse interpretieren. |
| Inhalt: | <ul style="list-style-type: none"> - Einführung in die Regelungstechnik - Erstellung mathematischer Modelle - Verhalten linearer Modelle - Übertragungsfunktionen - Stabilität - Sprungantwort linearer Systeme - Prinzip des Regelkreises - Wurzelortskurvenverfahren - Frequenzkennlinienverfahren - Nyquist-Diagramm - Erweiterte Regelkreisstrukturen - Experimentelle Modellbildung und Modellvereinfachungen - Heuristische Einstellregeln |
| Studien-/Prüfungsleistungen: | Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Grundlagen der Stochastik |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 180 Stunden, davon 55 Stunden Präsenz |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker |
| Angestrebte Lernergebnisse | Kenntnisse und Verständnis grundlegender stochastischer Methoden und Modelle, Einsatz in einfachen technischen Anwendungen. |
| Inhalt: | Teil 1: Beschreibende Statistik, Wahrscheinlichkeit, Simulation und Zufallszahlen, Zufallsvariable, Erwartungswert und Varianz, Spezielle diskrete Verteilungen, Ungleichung von Tschebyscheff, Schwaches Gesetz der großen Zahlen, Allgemeine Wahrscheinlichkeitsräume, Wahrscheinlichkeitsmaße auf $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, Schätzen, Testen Beispiele für technische Anwendungen Teil 2: Grundlagen der Mustererkennung, Parametrische und nicht-parametrische Dichteschätzungen, Mischmodelle und Expectation Maximization mit Spezialfall k-means Clustering |
| Studien-/Prüfungsleistungen: | Zu Teil 1: Klausur (120 min.) oder mündliche Prüfung (ca. 20 min.) wird durch Dozenten festgelegt und zu Veranstaltungsbeginn bekanntgegeben. Zu Teil 2: Kolloquium (ca. 20 min) nach bestandenem Teil 1 |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Immersive 3D Welten |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | IMW |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 4 SWS: 2 SWS Vorlesung 2 SWS Übung |
| Arbeitsaufwand: | 180: 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | C++, Graphische Simulation oder Computergraphik |
| Angestrebte Lernergebnisse | Der die Lernende kann: <ul style="list-style-type: none"> - gängige VR-Systeme (Software & Hardware) erklären - Unity 3D für VR-Systeme anwenden - State of the art-Entwicklungen zuordnen - Mensch-Computer Interaktionen vergleichen - Erstellung von 3D Welten beschreiben |
| Inhalt: | <ul style="list-style-type: none"> - Grundkonzepte 3D / Stereo - VR-Hardware, - VR-Eingabesysteme, - State of the art-Entwicklungen - Anwendungsgebiete - VR-Trends - Mensch-Computer Interaktion mit Sensoren |
| Studien-/Prüfungsleistungen: | Studienleistung: Übungsaufgaben in moodle Prüfungsleistung: Hausarbeit (ca. 15-20 Seiten exklusive Pflichtenheft & Bilder) |

| | |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Industrielle Netzwerke |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Sem. |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Elektrotechnik (Bachelor), Informatik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung/2 SWS ; Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden); 2 SWS Übung (30 Std.) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Lineare Algebra, Analysis für Informatiker, Einführung in die Programmierung für Informatiker, Einführung in C, Elektrotechnik für Informatiker |
| Angestrebte Lernergebnisse | Erarbeiten des Aufbaus und Wirkungsweise unterschiedlicher Netzwerke. Klassifizieren von Protokollen unterschiedlicher Netzwerke. Berechnung der Bitfehler- und Restfehlerraten in unterschiedlichen Netzwerke |
| Inhalt: | Klassen von Rechnernetzen, ISO-Schichtenmodell, Übertragungs- und Buszugriffstechniken, Netzwerkarten und Aufbau unterschiedlicher Netzwerkstopologien. Codierungsmöglichkeiten, Sicherungsverfahren, Berechnung von Bitfehlerraten- und Restfehler. |
| Studien-/Prüfungsleistungen: | Studienleistungen: Hausarbeit Prüfungsleistungen: Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Informatik und Gesellschaft |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Claude Draude |
| Sprache: | Englisch |
| Zuordnung zum Curriculum | Informatik Bachelor |
| Lehrform/SWS: | 2 SWS Seminar |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Aufgeschlossenheit für die Thematik nach dem 3. Semester des Grundstudiums |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> - Befähigung zu Eigenständigem Recherchieren und Einarbeiten in ein Thema - Befähigung zu methodisch-zielgerichteter Bearbeitung einer Problemstellung - Befähigung zu klaren Begrifflichkeiten, verständlicher und präziser Darstellung und fundierter Argumentation - Üben von Präsentationstechniken - Erweiterung des Verständnisses für gesellschaftliche Voraussetzungen und Wirkungen des wachsenden Einsatzes von Informationstechnik - Förderung der Fähigkeit, diese Wirkungen zu analysieren und zu bewerten |
| Inhalt: | <p>In his recent talk the founder of the World Economic Forum Klaus Schwab noted that we are living in an era of the Fourth Industrial Revolution that changes all aspects of life, from natural environment, to how we understand what it is to be human (Schwab 2016). The idea that technological change is both influenced by and in turn influences society and culture is not new. However, the widespread digitalization and the increasing scope, velocity and systemic impact of new technologies has made it even more urgent to think through the social conditions and socio-cultural implications of information technologies today. How exactly does society and computing influence each other? What are the most acute consequences and effects of digitalization? How can computing help to solve societal challenges? What are the ethical implications of technological developments that computer scientists should be aware of?</p> <p>Through the reading and discussion of selected texts, presentations and individual research, students will learn how to interpret, analyse and present interdisciplinary re-</p> |

| | |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>search on the relations between social factors and computing. They will also be encouraged to improve their writing and presentation skills, and to reflect on their own work in the broader social context.</p> <p>The seminar will be structured around several key themes:</p> <ul style="list-style-type: none"> - Social aspects of computing and digitalization of society; - Artificial Intelligence and social implications; - Development and the 'digital divide'; - Big data and politics; - Work, economy and the digital labor; - Social and ethical responsibility in computing. |
| Studien-/Prüfungsleistungen: | Participation in the classroom, writing a short paper, presentation |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Informationswirtschaft |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Jan Marco Leimeister |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor und andere |
| Lehrform/SWS: | 4 SWS Vorlesung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Informationswissenschaften I |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> - Kenntnisse über Informationswirtschaft: Aufgaben, Konzepte und praktische Lösungen - Kenntnisse im Bereich des Managements von Informationssystemen, strategisches Management von IT in Organisationen; Wissensmanagement, IT Governance, Führungsaufgaben im Informationsmanagement. |
| Inhalt: | <ul style="list-style-type: none"> - Aufgaben und Konzepte der Informationswirtschaft - Case Studies - Management von Informationssystemen - Strategisches IT-Management - Wissensmanagement - IT Governance - Führungsaufgaben im Informationsmanagement. |
| Studien-/Prüfungsleistungen: | Klausur (2 Std.) oder Hausarbeit (20 S.) oder Referat (20 Min.) mit schriftl. Ausarbeitung (ca. 12 S.) oder mündliche Prüfung (30 Minuten) |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Informationswissenschaften I |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Jan Marco Leimeister |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor und andere |
| Lehrform/SWS: | 4 SWS Vorlesung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> - Verständnis und Gestaltung von Informationssystemen in der Wirtschaft, Gesellschaft und Verwaltung im Anwendungszusammenhang - Verdeutlichung von Einführungs-, Nutzungs-, und Wartungsaspekten von Informationssystemen und deren unternehmensstrategische Möglichkeiten - Grundlegende Modellierungsfähigkeiten |
| Inhalt: | <ul style="list-style-type: none"> - Ziele und Aufgaben der Wirtschaftsinformatik - Technische Grundlagen moderner IT-Systeme, Netzwerksysteme und des Internets - Grundtypen von Modellen und Modellierung einfacher Sachverhalte - Modellierung im Kontext von Informationssystemen, Geschäftsprozessen - Kommunikations- und Informationstechnologie (IKT) und deren Rolle für die Gestaltung betrieblicher Abläufe und für die strategische Position der Unternehmen - Verschiedene Anwendungssysteme in betrieblichen Kontexten - Kennenlernen und Anwenden von zentralen Softwaretypen (Textverarbeitung, Tabellenkalkulation, Präsentation/Layout, Datenbanken) im Rahmen von Tutorien |
| Studien-/Prüfungsleistungen: | Klausur (2 Std.) oder mündliche Prüfung (30 Minuten) |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Intelligent Humanoid Robots I |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Robotik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor), |
| Lehrform/SWS: | 2 SWS oder 4 SWS Praktikum mit Hands-On Training |
| Arbeitsaufwand: | 90h oder 180h: 30h oder 60h Präsenzzeit 60h oder 120h Selbststudium (Projekt) |
| Kreditpunkte: | 3CP oder 6CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse in Java und/oder C/C++ |
| Angestrebte Lernergebnisse | Der/die Lernende kann: <ul style="list-style-type: none"> - die Anwendungen und Grenzen von humanoiden Robotern einschätzen - die Programmierung der humanoiden Roboter „NAO“ vornehmen - Python Skripte für NAO schreiben + die Naoqi-API bedienen - eigenständig kleinere Projekte mit dem NAO umsetzen Zusätzlich bei 6CP: <ul style="list-style-type: none"> - alle wichtigen Bibliotheken des NAOs kennen und anwenden - eigenständig größere Projekte mit dem NAO umsetzen |
| Inhalt: | <ul style="list-style-type: none"> - Grundlagen der Interaktion mit humanoiden Robotern - Vorstellen des NAO Programmiersystems & der Programmierung mit „Choreographie“ - Grundlagen / Prinzipien der Programmiersprache Python + Anwendung im NAO - Erweiterung der Funktionalität des NAOs Zusätzlich bei 6 CP: <ul style="list-style-type: none"> - Kooperation Roboter-Roboter / Mensch-Roboter - erweiterte Vision-Fähigkeiten oder komplexe motorische Anwendungen mit hohem Grad der Interaktion |
| Studien-/Prüfungsleistungen: | Studienleistungen: regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistungen bei 3 CP: Abschlussaufgabe mit Ausarbeitung Prüfungsleistungen bei 6 CP: umfangreiche Abschlussaufgabe mit Ausarbeitung und Präsentation |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Intelligente Technische Systeme |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Prozessor- und Rechner-technologie, Anwendungsgebiet Robotik |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Rechnerarchitektur, Lineare Algebra, Analysis für Informatiker |
| Angestrebte Lernergebnisse | Kenntnisse: Grundkenntnisse aus dem Bereich der Datenerfassung, Datenvorverarbeitung, Berechnung von Attributen, Techniken aus dem Bereich des Maschinellen Lernens Fertigkeiten: praktischer Einsatz verschiedener Techniken Kompetenzen: selbständige Entwicklung von einfachen Anwendungen |
| Inhalt: | Die Vorlesung beschäftigt sich hauptsächlich mit wesentlichen Grundlagen in verschiedenen Bereichen wie Sensorsysteme, Systemeigenschaften, grundlegende Signalverarbeitungsverfahren (digitale Filter, schnelle Fouriertransformation), Merkmalsselektionsverfahren (Filter und Wrapper, Principal Component Analysis), Grundlagen des maschinellen Lernens (Über- und Unteranpassung, Bias/Varianz-Problem, Techniken zur Evaluation wie Bootstrapping und Kreuzvalidierung, Evaluationsmaße), einfache Clustering- und Klassifikationsverfahren (c-means, hierarchische Verfahren, Naiver Bayes-Klassifikator, Nearest Neighbor Klassifikator) |
| Studien-/Prüfungsleistungen: | Schriftliche Prüfung (120 min.) oder mündliche Prüfung (ca. 20 min.) |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Internet-Suchmaschinen |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Knowledge & Data Engineering |
| ggf. Kürzel | IR |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatiker Mathematiker |
| Lehrform/SWS: | Vorlesung + Übung, je 2 SWS |
| Arbeitsaufwand: | 180 Stunden, davon 60 Stunden Präsenz |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Informatik Grundstudium |
| Angestrebte Lernergebnisse | Die Studierenden sollen in die Lage versetzt werden, methodische und analytische Ansätze aus dem Bereich des Information Retrieval anzuwenden und die Vor- und Nachteile der verschiedenen Verfahren bewerten zu können. |
| Inhalt: | Die Vorlesung gibt eine Einführung in das Gebiet des Information Retrievals. Unter IR versteht man im Allgemeinen das Finden von Informationen, wobei man dies häufig auf das Finden von Dokumenten, die die relevanten Informationen beinhalten, beschränkt. In der Vorlesung werden unter anderem neben den inhaltlichen Konzepten, die hinter bekannten Suchmaschinen wie z.B. Google oder Retrievalsystemen im Allgemeinen stehen, auch Ideen der effizienten Implementierung solcher Systeme eingeführt. Text im Modulhandbuch: Konzept, Methoden und Modelle zum Suchen und Finden von Informationen/ Dokumenten in großen Dokumentenbeständen; Architekturen und Anwendungen von IR-Systemen sowie die effiziente Umsetzung der eingeführten Modelle. |
| Studien-/Prüfungsleistungen: | Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Java Code–Camp Context Awareness I |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 4 SWS Vorlesung, Übung |
| Arbeitsaufwand: | 180 Stunden, davon 60 Stunden Präsenz |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, möglichst in Java |
| Angestrebte Lernergebnisse | Die Studierenden erhalten exemplarisch Einblick in die Programmierung mit Java für Sensoren und mobile Geräte (JME). Die Veranstaltung soll die Studierenden in die Lage versetzen, eigenständig kontextsensitive Anwendungen erarbeiten und erläutern zu können. |
| Inhalt: | Die Veranstaltung beinhaltet die Programmierung von Phidget-Sensoren und SunSPOTS sowie die Programmierung mit Java ME |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Knowledge Discovery |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Basis Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Robotik |
| ggf. Kürzel | KDD |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | Vorlesung + Übung, je 2 SWS |
| Arbeitsaufwand: | 180 Stunden, davon 60 Stunden Präsenz |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Informatik Grundstudium |
| Angestrebte Lernergebnisse | Die Studierenden besitzen einen Überblick über den Gesamtprozess der Wissensentdeckung und kennen die wichtigsten Methoden des überwachten und des unüberwachten Lernens. Sie sind in der Lage, die Vor- und Nachteile der verschiedenen Verfahren bewerten zu können, und die Verfahren im jeweiligen Kontext einzusetzen. |
| Inhalt: | Die Vorlesung gibt einen Überblick über Verfahren zur Wissensgewinnung aus strukturierten Daten und Texten. Behandelt werden <ul style="list-style-type: none"> • Techniken zur Vorverarbeitung und Integration von Datenbeständen, wozu das Konzept des Data Warehouse gehört. • OLAP-Techniken für die interaktive Analyse großer Datenbestände, • (halb-)automatische Verfahren zur Gewinnung neuen Wissens aus strukturierten Daten und Methoden zur Wissensextraktion aus Texten. Der Schwerpunkt der Vorlesung liegt auf den maschinellen Lernverfahren, deren Anwendung an konkreten Beispielen aufgezeigt wird. Die Vorlesung kann bei Interesse durch die Teilnahme am im folgenden Semester angebotenen Data Mining Cup (Projektseminar, 4 SWS) ergänzt werden. Kenntnis des Wissensentdeckungsprozesses und der eingesetzten Techniken. Hierzu gehören u.a. Entscheidungsbäume, Induktive Logikprogrammierung, Neuronale Netze, Clusteranalyse, Formale Begriffsanalyse, Assoziationsregeln. |
| Studien-/Prüfungsleistungen: | Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Komplexitätstheorie |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Kürzel | |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. Friedrich Otto |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor), Informatik (Master), Mathematik (Master) |
| Lehrform / SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | Die Studierenden verstehen die Grundlagen der Komplexitätstheorie. Sie verfügen über die Fähigkeit zur Anwendung in der Informatik. |
| Inhalt: | Rechnermodelle: Turingmaschinen, RAM etc. Komplexitätsmaße: Zeit und Platz Komplexitätsklassen: P, NP, PSPACE etc. Hierarchiesätze, untere Schranken, Reduzierbarkeit, vollständige Probleme |
| Studien-/Prüfungsleistungen: | Prüfungsleistung: Mündliche Prüfung (ca. 30 Min) Studienleistung: regelmäßige Bearbeitung der Übungsaufgaben |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Laborpraktikum Mobile Telefonsysteme |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | Laborpraktikum 2 SWS |
| Arbeitsaufwand: | 90 Stunden, davon 30 Stunden Präsenz |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Erfolgreiche Teilnahme an „Mobile Computing“ |
| Angestrebte Lernergebnisse | Die Studierenden erhalten Einblick in die praktische Umsetzung von Themen aus dem Gebiet der drahtlosen Kommunikationstechnik. Die Veranstaltung soll die Studierenden in die Lage versetzen, eigenständig praktische Aspekte der Kommunikationstechnik zu erarbeiten und erläutern zu können. |
| Inhalt: | Themen aus dem Bereich der drahtlosen Kommunikationstechnik, u.a.: <ul style="list-style-type: none"> • GSM • UMTS • Wlan • Bluetooth |
| Studien-/Prüfungsleistungen: | Ausarbeitung/mdl. Prüfung |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Laborpraktikum Rechnernetze |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | Laborpraktikum 2 SWS |
| Arbeitsaufwand: | 90 Stunden, davon 30 Stunden Präsenz |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Erfolgreiche Teilnahme an „Rechnernetze“ |
| Angestrebte Lernergebnisse | Die Studierenden erhalten Einblick in die praktische Umsetzung von Themen aus dem Gebiet der kabelgebundenen Kommunikationstechnik. Die Veranstaltung soll die Studierenden in die Lage versetzen, eigenständig praktische Aspekte der Kommunikationstechnik zu erarbeiten und erläutern zu können. |
| Inhalt: | Themen aus dem Bereich der kabelgebundenen Kommunikationstechnik, u.a.: <ul style="list-style-type: none"> • Verkabelung • Routing • Firewalls • VoIP |
| Studien-/Prüfungsleistungen: | Ausarbeitung/mdl. Prüfung |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Life Cycle Engineering |
| ggf. Modulniveau | Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | LCE |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. J. Hesselbach |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2SWS Vorlesung |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundkenntnisse der Technik, Mathematik und Chemie |
| Angestrebte Lernergebnisse: | Verständnis der Grundlagen der Umweltwirkungen durch die Herstellung, Nutzung und Entsorgung von Produkten. Kompetenzen bei der Analyse der Umweltwirkungen in allen Phasen des Produktlebenszyklus. Kenntnisse über die Vorgehensweise bei der Erstellung, Bewertung und Nutzung von Umweltbilanzen. Übersicht der softwaretechnischen Anwendungen zur Erstellung von Ökobilanzen Grundlagen der softwaretechnischen Umsetzung von Ökobilanzen für einfache Produkte |
| Inhalt: | 1. Übersicht bezüglich Umweltwirkungen (Ozonloch, Treibhauseffekt, Photosmog, Ressourcenverknappung, Waldsterben, Überdüngung, Toxizität) 2. Staatliche und betriebliche Instrumente zur Umsetzung von Umweltschutzmaßnahmen 3. Life Cycle Engineering. Vorgehensweise bei Erstellung von Ökobilanzen 4. Ausgewählte Beispiele von Ökobilanzen 5. Handlungsmöglichkeiten zum Schutz der Umwelt 6. Softwaresysteme zur Erstellung von Umweltbilanzen |
| Studien-/Prüfungsleistungen: | Klausur, 60 Minuten |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Life Cycle Engineering Praktikum |
| ggf. Modulniveau | Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | LCE P |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. J. Hesselbach |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2SWS Praktikum |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Um an diesem Praktikum teilnehmen zu dürfen, müssen Sie Klausur zur Lehrveranstaltung Life Cycle Engineering bestanden haben. |
| Angestrebte Lernergebnisse: | Praktische Anwendung der in LCE erlernten Inhalte |
| Inhalt: | Zerlegen eines Produktes Aufschlüsseln der Bauteile Abbildung des Produktes in einer Bilanzierungssoftware Erstellung einer Life Cycle Bilanz für das Produkt |
| Studien-/Prüfungsleistungen: | Ausarbeitung der Praktikumsresultate (Abschlussbericht) mit Abschlusspräsentation (20 Minuten) |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Managing IT-enabled Change |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Dr. Dirk Basten |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik(Bachelor) und andere |
| Lehrform/SWS: | 4 SWS (Seminar) |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden (Vorlesung) 30 Stunden (Seminar) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | Die Studierenden haben ein breites und integriertes Wissen arbeitswissenschaftlicher Grundlagen und sind in der Lage ihr Wissen selbstständig zu vertiefen. Studierende... ...erklären die organisatorischen Voraussetzungen für die Einführung eines Softwaresystems. ... erklären die Probleme des organisatorischen Wandels. ...beschreiben die Ansätze zur Erklärung von Technologieadoption. ...erklären die Ursachen von Widerstand. ...wenden die Methoden des Change Managements zur Bewältigung des organisatorischen Wandels an. |
| Inhalt: | <ul style="list-style-type: none"> • Geplanter organisatorischer Wandel • IT-basierte Veränderung • Modelle zu Technologieakzeptanz und Verbreitung von Innovationen • Abstimmung von Technologie und Unternehmensstrategie, -kultur, -prozessen • Widerstand: Ursachen und Gegenstrategien <p>Einführung: Erarbeitung von Motivation und grundlegenden Begriffen des organisationalen Wandels. Theorie: Vorstellung und Diskussion von systematischen Vorgehensweisen und Theorie des organisationalen Wandels. Fallstudien: Fallstudien zu Vorgehen und Auswirkungen von IT-basiertem Wandel. Anwendung: Theoriebasierte Analyse der Fallstudien und Entwicklung von verbesserten Vorgehensweisen.</p> |
| Studien-/Prüfungsleistungen: | Schriftliche (60 min.) oder mündliche Prüfung (20 min.) (nach Teilnehmerzahl) Seminar: Seminarvortrag (Fallstudienpräsentation) |

| | |
|--|------------------------------------------------------------------------------------------------------------------------------------------|
| | Die Bewertung erfolgt auf Basis der mündlichen Beteiligung (20%), der Präsentation von Fallstudien (30%) und der Abschlussprüfung (50%). |
|--|------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mathematische Modelle zur MTTF und PFD Analyse |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechnertechnologie |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik(Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung/2 SWS ; Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Lineare Algebra, Analysis für Informatiker, Einführung in die Programmierung für Informatik, Einführung in C, Elektrotechnik für Informatiker |
| Angestrebte Lernergebnisse | Erarbeiten und erschließen der Mathematischen Modelle zur Berechnung von MTTF und PFD von unterschiedlichen Rechnerarchitekturen. Verallgemeinern der Grundlagen und mathematische Modelle für unterschiedliche Systemstrukturen. |
| Inhalt: | Vorstellung der Mathematische Modelle zur MTTF und PFD Analyse, Wahrscheinlichkeitstheorie. Erarbeitung der mathematischen Modellbeschreibungen, Markov-Modelle, McLaurin- und Tayler-Reihen, DGL im Zeitbereich zur Berechnung der Ausfallwahrscheinlichkeiten |
| Studien-/Prüfungsleistungen: | Studienleistungen: Hausarbeit, Präsentation, Projektarbeit Prüfungsleistungen: Klausur oder mündliche. Prüfung |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mensch-Maschine-Systeme 1 |
| ggf. Modulniveau | Basis Anwendungsgebiet Computergraphik, Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Computergrafik, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | MMS 1 |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Ludger Schmidt |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Maschinenbau, Mechatronik, Informatik, Wirtschaftsingenieurwesen Maschinenbau |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Seminar |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden (Vorlesung) 30 Stunden (Seminar) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Angestrebte Lernergebnisse | Die Studierenden haben ein breites und integriertes Wissen und Verstehen der Grundlagen für die Analyse, den Entwurf und die Bewertung von Mensch-Maschine-Systemen |
| Inhalt: | Technologisch-technische Gestaltung Ergonomische Gestaltung und Anthropometrie Menschliche Informationsverarbeitung und informationstechnische Gestaltung Regler-Mensch-Modell Cognitive Engineering und menschliche Fehler |
| Studien-/Prüfungsleistungen: | Vorlesung: Schriftliche (90 min.) oder mündliche (20 min.) Prüfung (nach Teilnehmerzahl) Seminar: Seminarvortrag oder Hausarbeit |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mensch-Maschine-Systeme 2 |
| ggf. Modulniveau | Basis Anwendungsgebiet Computergraphik, Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Computergrafik, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | MMS 2 |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Univ.-Prof. Dr.-Ing. Ludger Schmidt |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Maschinenbau, Mechatronik, Informatik, Wirtschaftsingenieurwesen Maschinenbau |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Seminar |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden (Vorlesung) 30 Stunden (Seminar) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Mensch-Maschine-Systeme 1 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Methoden für die Mensch-Maschine-Systemgestaltung und sind in der Lage ihr Wissen selbstständig zu vertiefen. |
| Inhalt: | Benutzerorientierter Gestaltungsprozess und Analyse des Nutzungskontextes Aufgabenanalyse Design-Methoden Normen und Richtlinien bei der prototypischen Gestaltung User Interface Design Patterns Prototypische Entwicklung am Beispiel Mensch-Roboter- Interaktion Evaluationsmethoden Statistische Methoden Planung, Durchführung und Auswertung experimenteller Untersuchungen Fallbeispiel für experimentelle Untersuchungen |
| Studien-/Prüfungsleistungen: | Vorlesung: Schriftliche (90 min.) oder mündliche (20 min.) Prüfung (nach Teilnehmerzahl) Seminar: Seminarvortrag oder Hausarbeit |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mikroprozessortechnik – Labor |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Robotik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Labor 2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 4 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Mikroprozessortechnik und eingebettete Systeme 1, Programmierkenntnisse |
| Angestrebte Lernergebnisse | Grundlagen, Funktionsprinzipien und Systemarchitekturen von einfachen Mikroprozessoren erlernen sowie marktübliche Ausprägungen kennenlernen. Darstellung von Informationen für Mikroprozessoren, Aufbau und Wirkungsweise von Rechenwerken, Leitwerk und ALUs; Grundlegender Aufbau eines Mikroprozessors, Systembusschnittstelle, Zeitverhalten, Adressdekodierung, Adressierungstechniken. Entwurf von Mikroprozessor basierenden Systemen erlernen (insbesondere Design, Modellierung und Implementierung) |
| Inhalt: | Aufbau und Programmierung von Mikrocontrollern auf Basis von-Neumann Rechnern (z.B. MC6809, MPC430 o.ä.). Programmierung durch höhere Programmiersprache (z.B. C). Lösen von typische Anforderungen aus den Bereich Mikroprozessortechnik. Programmiertechniken |
| Studien-/Prüfungsleistungen: | Studienleistung: Hausarbeit, Bericht, Projektarbeit Prüfungsleistung: Mündliche Prüfung |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mikroprozessortechnik und eingebettete Systeme 1 |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Prozessor- und Rechnertechnologie |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Programmierkenntnisse |
| Angestrebte Lernergebnisse | Erarbeiten der Grundlagen, Funktionsprinzipien und Systemarchitekturen von einfachen Mikroprozessoren sowie marktübliche Ausprägungen kennenlernen. Aufstellen der Darstellung von Informationen für Mikroprozessoren. Beschreiben des Aufbaus und Wirkungsweise von Rechenwerken, Leitwerk und ALUs. Herausstellen des grundlegenden Aufbaus eines Mikroprozessors, Systembusschnittstelle, Zeitverhalten, Adressdekodierung, Adressierungstechniken. Entwurf von Mikroprozessor basierenden Systemen erlernen (insbesondere Design, Modellierung und Implementierung) |
| Inhalt: | Vorstellung der Technologie, der Funktionsweise und der Architektur von Mikroprozessoren. Typische Anforderungen und Beispiele werden vorgestellt. Modellierung von Mikroprozessor-Systemen (Hard- und Software). Echtzeitaspekte und Verteilungaspekte, Betriebssysteme und Programmierstechniken |
| Studien-/Prüfungsleistungen: | Studienleistungen: Hausarbeit Prüfungsleistungen: Mündliche Prüfung oder Klausur |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mikroprozessortechnik und eingebettete Systeme 2 |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechnertechnologie |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Sem. |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Elektrotechnik (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | Vorlesung/2 SWS Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 ECTS |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Programmierkenntnisse, abgeschlossenes Grundstudium |
| Angestrebte Lernergebnisse | Grundlagen, Funktionsprinzipien und Systemarchitekturen von moderner Mikroprozessoren- Technologie übliche Ausprägungen kennenlernen, anschließend den Entwurf von modernen 32 Bit Mikroprozessor basierenden Systemen erlernen. Hochleistungsmikroprozessoren. Aufbau von CISC, RISC und EPIC Mikroprozessoren. Wirkungsweise von CISC, RISC und EPIC-Mikroprozessoren. Beschleunigungsmaßnahmen zur Leistungssteigerung von Mikroprozessoren. Aufbau und Wirkungsweise von Pipeline. Fehlermodelle von Pipeline. Superskalare Mikroprozessoren. Vorteile von EPIC-Mikroprozessoren |
| Inhalt: | Vorstellung moderner Mikroprozessoren-Technologie, der Funktionsweise und der CISC und RISC Architekturen von Mikroprozessoren. Aufbau der Prozessor-Pipeline, Programmiermodell, Adressierungsarten, Instruction- Cache-Systeme und Steuerung. Teilbare und nicht teilbare Busoperationen. RISC-Architekturen, Fetch/Decode-Prinzipien, Super-Pipeline-Architekturen, Out-of-Order Execution, Branch-Prediction. Erweiterung der RISC Mikroprozessoren auf EPIC durch Speculation- und Predecation Mechanismen. Typische Anforderungen und Beispiele werden vorgestellt. Modellierung von Mikroprozessor-Systemen (Hard- und Software). Echtzeitaspekte und Programmiertechniken |
| Studien-/Prüfungsleistungen: | schriftliche Prüfung, mündliche Prüfung, Hausarbeit |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Mobile Computing |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Klaus David |
| Sprache: | Englisch |
| Zuordnung zum Curriculum | Informatiker/ Elektrotechniker |
| Lehrform/SWS: | Vorlesung, Übung, 4 SWS |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium, |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Erfolgreiche Teilnahme an „Rechnernetze“ |
| Angestrebte Lernergebnisse | Kennenlernen der theoretischen Grundlagen, aktuellen Systemen und insbesondere Anwendungen der mobilen Kommunikation und deren Entwicklung |
| Inhalt: | <ul style="list-style-type: none"> • Mobilfunkkanal und Funkübertragung • GSM Dienste (Sprache, Daten, Sicherheitsfunktionen) • GSM System (BSS, MSC), GPRS, EDGE • UMTS (HSUPA/HSDPA) • W-LAN • Dienste wie MMS, Webbrowsen, push email, location based services ... • Mobile Betriebssysteme • Software für Anwendungsentwicklung • pervasive computing, ubiquitous systems |
| Studien-/Prüfungsleistungen: | Klausur oder mündliche Prüfung |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Modellbildung und Simulation: Lokale und regionale Umweltprobleme |
| ggf. Modulniveau | Anwendungsgebiet Umweltinformatik |
| ggf. Kürzel | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. Joseph Alcamo PD Dr. Rüdiger Schaldach |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2SWS Seminar |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundlagen der Umweltinformatik |
| Angestrebte Lernergebnisse: | Kenntnisse über grundlegende Methoden der Modellbildung und Simulation in den Umweltwissenschaften. |
| Inhalt: | Grundlagen der Systemtheorie; Schritte der Modellbildung; Modellkonzepte und deren Eignung für Problemstellungen in den Umweltwissenschaften; Methoden zur Evaluierung von Modellen; Entwurf und Durchführung von Simulationsstudien. |
| Studien-/Prüfungsleistungen: | Seminarvortrag oder Klausur |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bezeichnung: | Nutzungsorientierte Gestaltung |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt, Anwendungsgebiet Software Tools Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. phil. Claude Draude |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor |
| Lehrform/SWS: | 4 SWS: 2 SWS Vorlesung/Seminar 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 60 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> - Grundlagen nutzungsorientierter Gestaltung - Kontext und Bedarfe technischer Entwicklungen erheben - Erprobung partizipativer Methoden - Wissenschaftliches Arbeiten in Form von forschendem Lernen |
| Inhalt: | <p>Technische Entwicklungen sollen benutzbar sein und in verschiedenen Arbeits- und Lebenskontexten funktionieren. Nutzungsorientierte Gestaltung hilft, Fehlentwicklungen und mangelnde Akzeptanz zu vermeiden. Soziale und organisatorische Aspekte, sowie Bedarfe unterschiedlicher Personen und Zielgruppen zu erheben und für technische Anforderungen aufzuarbeiten, ist ebenso notwendig, wie herausfordernd.</p> <p>Die Veranstaltung stellt verschiedene Ansätze nutzungsorientierter Gestaltung, wie z.B. Partizipatives Design, User-Centered Design oder Design for All vor. In der Übung werden verschiedene Methoden, wie z.B. das Arbeiten mit Interviews im Kontext, Personas, Szenarien oder Cultural Probes, experimentell und empirisch erprobt.</p> |
| Studien-/Prüfungsleistungen: | Anwendung einer nutzungsorientierten Methode mit kommentierter, schriftlicher Ausarbeitung. |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bezeichnung: | Parallelverarbeitung I |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Software Tools |
| ggf. Kürzel | PV1 |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Claudia Fohry |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor u. andere |
| Lehrform/SWS: | Ca. 1,5 SWS Vorlesung, ca. 0,5 SWS praktische Übung in Projektform |
| Arbeitsaufwand: | 30 Stunden Präsenz 60 Stunden Selbststudium, |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Einfprog |
| Angestrebte Lernergebnisse | Kenntnis grundlegender Konzepte der Parallelverarbeitung, Fähigkeit zum Denken in parallelen Algorithmen, Verständnis für Effizienzüberlegungen, Fertigkeit in Entwicklung paralleler Algorithmen, Kenntnis eines parallelen Programmiersystems für Rechner mit gemeinsamen Speicher, Fertigkeiten in Entwicklung von Programmen mit diesem System, Kenntnis von Anwendung der Parallelverarbeitung, vertiefte Programmierfertigkeiten, Entwicklung von Fähigkeit zur Teamarbeit und Projektorganisation |
| Inhalt: | Grundbegriffe, Beispielalgorithmen und -anwendungen, Programmierung am Beispiel eines Programmiersystems (z. B. OpenMP, X10), Entwurfsmuster, Effizienzkriterien |
| Studien-/Prüfungsleistungen: | Projektarbeit |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Parallelverarbeitung II |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Software Tools |
| ggf. Kürzel | PV2 |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Claudia Fohry |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor u. andere |
| Lehrform/SWS: | Ca. 1,5 SWS Vorlesung, ca. 0,5 SWS praktische Übung in Projekform |
| Arbeitsaufwand: | 30 Stunden Präsenz 60 Stunden Selbststudium, |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | PV1 |
| Angestrebte Lernergebnisse | Kenntnis zweier weiterer Programmiersysteme (z.B. MPI, Cuda, X10 für verteilten Speicher), vertiefte Fertigkeiten in Entwicklung paralleler Algorithmen und Umsetzung mit diesen Programmiersystemen unter Berücksichtigung der Effizienz, Kennenlernen von Beispielanwendungen, Entwicklung von Fähigkeit zur Teamarbeit und Projektorganisation |
| Inhalt: | Parallele Programmierung mit weiteren Programmiersystemen (z. B. MPI, Cuda, X10 für verteilten Speicher), Beispielalgorithmen und -anwendungen |
| Studien-/Prüfungsleistungen: | Projektarbeit |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum Digitaltechnik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | nach Absprache |
| Zuordnung zum Curriculum | Elektrotechnik (Bachelor), Informatik (Bachelor) |
| Lehrform/SWS: | 2 SWS: 2 SWS Praktikum |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Vorbereitungszeit |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Vorlesung Digitale Logik, sicherer Umgang mit Messgeräten (optional, z. B. aus dem Elektrotechnischen Praktikum I und II bzw. Messtechnischen Praktikum) |
| Angestrebte Lernergebnisse | <ul style="list-style-type: none"> • Praktischer Umgang mit digitalen Schaltungen • Vertiefen der Fähigkeiten aus der Vorlesung Digitaltechnik 1 • Vertiefung des Verständnisses von Entwurf und Funktionsweise digitaler Schaltungen • Systematische Analyse (fehlerbehafteter) Schaltungen. |
| Inhalt: | <ul style="list-style-type: none"> • Gatterfunktionen • Kombinatorische Logik • Sequentielle Logik • Zustandsautomaten • FPGA-Programmierung |
| Studien-/Prüfungsleistungen: | mündl. Prüfung und Hausarbeit und Bericht (Versuchsausarbeitung) |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum Intelligente Eingebettete Systeme |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Prozessor- und Rechnertechnologie, Anwendungsgebiet Robotik |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 2 SWS Praktikum |
| Arbeitsaufwand: | 90 Stunden, davon 30 Stunden Präsenz |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker, Rechnerarchitektur, Einführung in C, Intelligente Technische Systeme |
| Angestrebte Lernergebnisse | Kenntnisse: Vertiefung der Kenntnisse der Vorlesung Intelligente Technische Systeme oder einer vergleichbaren Vorlesung Fertigkeiten: Auswahl und Einsatz von Techniken aus verschiedenen Bereichen Kompetenzen: selbständige Entwicklung einer komplexeren Anwendung im Team |
| Inhalt: | In diesem Praktikum werden Kenntnisse aus verschiedenen Bereichen vertieft und Techniken aus diesen Bereichen kombiniert, um eine komplexe Anwendung zu entwickeln. Dazu gehören hardwarenahe Programmierung in C, Methoden zur Datenerfassung und Datenvorverarbeitung, einfache Algorithmen zur Musterklassifikation, Methoden des maschinellen Lernens. Die Aufgabenstellung kann variieren. Verwendet werden beispielsweise Roboterbausätze des Typs Asuro (entwickelt am DLR) oder Sports Watch Sensor Systeme (z.B. Texas Instruments EZ430) |
| Studien-/Prüfungsleistungen: | Projektarbeit |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum Internet-Suchmaschinen |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Knowledge & Data Engineering |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mathematik |
| Lehrform/SWS: | 2 SWS Praktikum |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Praktikum (30 Stunden) Selbststudium: 60 Stunden |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Gleichzeitiger oder bereits abgeschlossener Besuch der Vorlesung Internet-Suchmaschinen |
| Angestrebte Lernergebnisse | |
| Inhalt: | Im Praktikum werden die in der Vorlesung Internet-Suchmaschinen vermittelten Konzepte in die Praxis umgesetzt. Aufgeteilt auf mehrere Schritte implementieren die Teilnehmer dabei ihre eigene Suchmaschine. |
| Studien-/Prüfungsleistungen: | Bearbeitung der Praktikumsaufgaben, mündliche Prüfung |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum Knowledge Discovery |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiete Knowledge & Data Engineering, Anwendungsgebiete Knowledge & Data Engineering |
| ggf. Kürzel | PKDD |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Gerd Stumme |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 2 SWS Praktikum |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Praktikum (30 Stunden) Selbststudium: 60 Stunden |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Knowledge Discovery |
| Angestrebte Lernergebnisse | Die Teilnehmer lernen, wie die in der Vorlesung Knowledge Discovery vorgestellten überwachten und unüberwachten Lernverfahren werden in die Praxis umgesetzt werden und wie große Datenmengen mit ihnen verarbeitet werden können. |
| Inhalt: | Die in der Vorlesung Knowledge Discovery vorgestellten überwachten und unüberwachten Lernverfahren werden in die Praxis umgesetzt |
| Studien-/Prüfungsleistungen: | Projektarbeit |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum kooperative verteilte Robotersysteme |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Robotik |
| ggf. Kürzel | PKVR |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Kurt Geihs |
| Sprache: | Deutsch und Englisch |
| Zuordnung zum Curriculum | Informatik (Bachelor) u. andere |
| Lehrform/SWS: | 4 SWS Praktikum |
| Arbeitsaufwand: | 60 Stunden Präsenz und Aufgabenbearbeitung 90 Stunden Vor- und Nachbereitung 30 Stunden Prüfungsvorbereitung |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse | <p>Ziel des Praktikums ist es, den Studierenden die Möglichkeit zu bieten, die im bisherigen Studium erworbenen theoretischen Kenntnisse und Fähigkeiten in der Bearbeitung aufeinander aufbauender Aufgabenstellungen unter Anleitung praktisch anzuwenden und zu erweitern. Dafür steht den Teilnehmern eine netzwerkfähige und mobile Roboterplattform zur Verfügung (Surveyor SRV-1 Blackfin, siehe http://www.surveyor.com).</p> <p>Die Teilnehmer erlernen grundlegende Techniken der Robotik an einfachen praktischen Beispielen. Dies beinhaltet hardwarenahe Programmierung, Roboterarchitektur, Sensorik, Computervision, einfache Methoden der Roboterentscheidungsfindung und Roboteraktik (z.B. Pfadplanung) bis hin zur kooperativen verteilten Steuerung von Robotersystemen. Neben den roboterspezifischen Grundlagen werden den Teilnehmern auch praxisrelevante Inhalte vermittelt. Hierzu zählen Grundfertigkeiten der Hardware- und Softwareentwicklung, wie beispielsweise Löten, Sensorintegration, Netzwerkkommunikation, GUI Programmierung, Cross Compiling und das Anwenden eines Versionierungssystems.</p> <p>In der Regel finden die Versuche als Gruppenarbeit statt.</p> |
| Inhalt: | <p>Kennenlernen einer Roboterarchitektur, Programmierung der Roboterplattform, Sensordatenverarbeitung, Entscheidungsfindung in Robotersystemen, Steuerung der Aktorik, Einblicke in verteilte kooperative Robotersysteme. Am Ende des Praktikums findet eine Roboter-Challenge statt, in der die erlernten Kenntnisse frei und kreativ eingesetzt werden sollen, um eine umfangreiche, allgemein beschriebene Aufgabe</p> |

| | |
|------------------------------|------------------------------------------------------------------------------------------------------------------|
| | selbstständig zu lösen. Die erarbeiteten Lösungen der einzelnen Gruppen werden im direkten Wettbewerb evaluiert. |
| Studien-/Prüfungsleistungen: | Projektarbeit/mündliche Prüfung/schriftliche Prüfung |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum Mensch-Maschine-Interaktion |
| ggf. Modulniveau | Basis Anwendungsgebiet Computergraphik, Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Computergrafik, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | P-MMI |
| Studiensemester: | ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. Ludger Schmidt |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Maschinenbau, Mechatronik, Informatik, Wirtschaftsingenieurwesen Maschinenbau |
| Lehrform/SWS: | 2 SWS Praktikum |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden Selbststudium: 60 Stunden |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Mensch-Maschine-Systeme 1 und/oder 2 |
| Angestrebte Lernergebnisse | Vertiefte Wissensbestände hinsichtlich Mensch-Maschine-Interaktionsprinzipien werden von den Studierenden durch experimentell erfahrungsgelitetes Lernen erarbeitet. |
| Inhalt: | Visuelle Wahrnehmung: Sehschärfe, Farbsehen und räumliches Sehen Auditive Wahrnehmung: Richtungshören, Hörschwelle und Maskierungseffekte Haptische Wahrnehmung Vestibuläre Wahrnehmung Grundlagen der menschlichen Informationsverarbeitung Brain-Computer-Interface Manuelle Regelung einer kritischen Regelungsaufgabe Fehlermanagement Fahrer-Fahrzeug-Interaktion bei Nebenaufgaben Physiologische Belastungs- und Beanspruchungsanalyse |
| Studien-/Prüfungsleistungen: | Praktikumsbericht |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Praktikum VR Game Programmierung |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Computergraphik, Anwendungsgebiet Computergraphik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Dieter Wloka |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 2 SWS Praktikum |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Vorkenntnisse im Bereich 3D Modellierung/Game Engine/Unity. |
| Angestrebte Lernergebnisse | Fundierte Kenntnisse über die Grundlagen von Virtual Reality Anwendungen. Fertigkeit eine Virtual Reality Anwendung unter Vermeidung von Motion sickness zu planen, diese für die Endgeräte ausreichend performant zu entwickeln und entsprechend zu testen. Verständnis der allgemeinen Funktionsweise von aktuell verfügbaren Game-Engines und die Nutzung dieser mit State-of-the Art Entwicklungstools im Bereich Virtual Reality. |
| Inhalt: | Geplante Themen sind: Herausarbeitung von Unterschieden zu Desktop basierten 3D Anwendungen. Entwicklung von multi-user Anwendungen/Inhalten für Virtual Reality Plattformen. Betrachtung verschiedener Fortbewegungsmechanismen unter Vermeidung der Motion sickness. Betrachtung von Optimierungsaspekten, um VR Anwendungen performant auf den Endgeräten ausführen zu können. Die Veranstaltung „Praktikum 3D Game Programmierung“ unterscheidet sich inhaltlich und stellt keine Voraussetzung für dieses Praktikum dar. |
| Studien-/Prüfungsleistungen: | Projekt und Hausarbeit |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Programmierung mobiler Geräte (Android und iOS) |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Klaus David |
| Sprache: | Deutsch/Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 4 SWS: 2 SWS Vorlesung 2 SWS Übung |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit 120 h Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse in Java und objektorientierter Programmierung, Besuch der Lehrveranstaltungen „Einführung in die Programmierung für Informatik“, „Programmiermethodik“ |
| Angestrebte Lernergebnisse | Die Studierenden erhalten einen Einblick in die Programmierung mit Android und IOS. Die Veranstaltung soll die Studierenden in die Lage versetzen, eigenständig Apps für Smartphones zu erarbeiten und die Vorgehensweise dazu erläutern zu können. Zusätzlich wird das Programmieren im Team vertieft. |
| Inhalt: | Einführung in die Programmierung mit Android und IOS auf mobilen Geräten |
| Studien-/Prüfungsleistungen: | Form: Studienleistungen (b/nb): Bericht, Software-Implementierung, Anwesenheitspflicht 80% Prüfungsleistungen: mündliche Prüfung oder Klausur oder Praxisteil und Referat/Präsentation Dauer: mündlich 30, schriftlich 120 Minuten |

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Prozessleittechnik |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 4 SWS Vorlesung |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit 120 h Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, Grundlagen der Informatik, Digitaltechnik, Mikroprozessoren oder Rechnerarchitektur, Regelungstechnik, Mathematik |
| Angestrebte Lernergebnisse | Der/die Lernende kann: <ul style="list-style-type: none"> - Aufbau und Wirkungsweise von Prozessleitsystemen verstehen, deren Hard- und Softwarekomponenten bewerten, den Einsatz von PLT-Einrichtungen in der Verfahrens- und Anlagensicherheit erklären. - neue PLT-Einrichtungen und Anwendungen eigenständig planen und realisieren, existierende PLT-Einrichtungen, Verfahren und Anwendungen kritisch hinterfragen, vergleichen und bewerten. |
| Inhalt: | Die Vorlesung beschäftigt sich mit Strukturen von Prozessen, Mathematischen Modellbeschreibungen, Aufbau von Prozessleitsystemen – und Automatisierungssystemen, Aufbau und Wirkungsweise von Peripherieeinheiten, Echtzeiteigenschaften, Programmierung, Vorstellung marktüblicher Systeme und Werkzeuge mit Bezug auf die Anwendung, Beispielanwendungen aus verschiedenen Applikationen |
| Studien-/Prüfungsleistungen: | Je nach Teilnehmerzahl: Klausur 60 – 180 Minuten, mündliche Prüfung 20 – 40 Minuten |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Reduktionssysteme I |
| ggf. Modulniveau | Wahlpflicht Schwerpunkt |
| ggf. Kürzel | |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortlicher: | Prof. Dr. Friedrich Otto |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Mathematik (Master), Informatik (Bachelor), Informatik (Master) |
| Lehrform / SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Eigenstudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | <p>Allgemein: Die Veranstaltung behandelt Techniken zum Rechnen in durch Gleichungen definierte Strukturen.</p> <p>Kompetenzen: Das vermittelte Methodenwissen hilft den Studierenden einzuschätzen, ob und ggf. welche Reduktionstechniken eingesetzt werden können bei der Lösung algorithmischer Probleme in durch Gleichungen definierten Strukturen, wie sie beispielsweise bei der Implementierung funktionaler Sprachen, bei der Programmspezifikation, der automatischen Programmverifikation und der deklarativen Programmierung auftreten.</p> <p>Berufsvorbereitung: Die Veranstaltung bereitet auf den Einsatz in der Softwareentwicklung vor.</p> |
| Inhalt: | <ul style="list-style-type: none"> - Termersetzungssysteme - Gleichheitstheorien - Satz von Birkhoff - Reduktionsrelationen - Termination und Konfluenz - Knuth-Bendix Vervollständigung |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung (ca. 30 Min.) nach erfolgreicher Teilnahme an den Übungen |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Risikobewertung von Rechnerarchitekturen I |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechner-technologie |
| ggf. Kürzel | |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Maschinenbau (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden) 2 SWS Übung (30 Stunden) Selbststudium: 120 Std. |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik, Rechnerarchitektur, Mathematik |
| Angestrebte Lernergebnisse | Erarbeitung der Grundlagen der Risikobestimmung von unterschiedlichen Rechnerarchitekturen. Aufstellung und Bestimmung von Risikopotentialen in Hard- und Softwarekomponenten, Erarbeiten der Grundlagen zur mathematischen Modellierung und Beschreibungen |
| Inhalt: | Vorstellung und Anwendung von Risikoberechnung, Risikograph, Wahrscheinlichkeitstheorie. Kennlernen und Analyse von Strukturen unterschiedlichen Rechnerarchitekturen. Anwendung von mathematischen Modellbeschreibungen und Berechnung der Modelle |
| Studien-/Prüfungsleistungen: | Klausur, mündliche. Prüfung, Hausarbeit, Referat/Präsentation |

| | |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Sicherheit in Kommunikationsnetzen |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Kommunikation und Verteilte Systeme, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | SK |
| ggf. Lehrveranstaltungen | Vorlesung, Übung |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortliche(r): | Prof. Dr. Arno Wacker |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor u. andere |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 CP |
| Empfohlene Voraussetzungen: | Grundlagen der angewandten Kryptologie, Rechnernetze, Techniken und Dienste des Internets, Einführung in die Programmierung für Informatik |
| Angestrebte Lernergebnisse | Fundierte Kenntnisse über die Grundlagen von Kommunikationsprotokollen zur sicheren Nachrichtenübermittlung. Verständnis der allgemeinen Funktionsweise von aktuell eingesetzten sicheren Kommunikationsprotokollen und die Fähigkeit selbst aus kryptographischen Primitiven sichere Kommunikationsprotokolle abzuleiten. Fertigkeit um die Sicherheit von Kommunikationsprotokollen selbst zu analysieren und einzuschätzen. |
| Inhalt: | Im Rahmen dieser Vorlesung werden verschiedene Methoden zur sicheren Nachrichtenübertragung in modernen Kommunikationsnetzen (wie z.B. dem Internet) vorgestellt und analysiert. Dabei steht das Zusammenspiel der kryptographischen Algorithmen und deren (un-)sicheren Anwendung im Vordergrund und nicht die (hier als sicher angenommen) kryptographischen Algorithmen selbst. (Für das Verständnis dieser kryptographischen Algorithmen sei auf die Vorlesung "Grundlagen der angewandten Kryptologie" verwiesen.) Vorgestellt werden dabei Protokolle zur Authentifizierung (z.B. Needham-Schröder, Kerberos), Protokolle für den sicheren Datenaustausch im Internet (z.B. IPSec, SSL/TLS, SSH, PKI, SMIME), Protokolle für die Sicherung von drahtlosem Datenverkehr (z.B. WEP, WPA) und sichere Protokolle für den Datenaustausch in optischen Netzen (z.B. BB84 basierend auf Quantenkryptographie). Dabei werden sowohl die Funktionsweise als auch die aktuell |

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>bekanntesten Schwächen vorgestellt und analysiert (z.B. Needham-Schröder, WEP).</p> <p>Schwerpunkt: Bewertung der Sicherheit von aktuell eingesetzten sicheren Kommunikationsprotokollen, Steigerung des Sicherheitsbewusstseins beim täglichen Umgang mit modernen Kommunikationsnetzen, wie z.B. dem Internet.</p> |
| Studien-/Prüfungsleistungen: | Mündliche Prüfung |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Signalverarbeitung mit Mikroprozessoren I |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Prozessor- und Rechner-technologie |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Priv.-Doz. Dr. habil. Hans-Dieter Wacker |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik Bachelor u. andere |
| Lehrform/SWS: | Vorlesung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 4 SWS / 6 CP ¹ |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, Grundlagen der Informatik, Digitale Logik, Mikroprozessoren oder Rechnerarchitektur, Mathematik. |
| Angestrebte Lernergebnisse | Erarbeitung und erschließen von mathematische Grundlagen und Modelle von unterschiedlichen Methoden der digitalen Signalverarbeitung. Laplace- Transformation, Fourier-Transformation, z- Transformation. Verallgemeinern der erworbenen Kenntnisse auf digitale Filtersysteme |
| Inhalt: | Shannon-Theorem, Laplace-Transformation, Fourier- Transformation, z-Transformation, Berechnungen von FIR- und IIR-Filtern zur Signalanalyse, Beispielanwendungen aus verschiedenen Applikationen. |
| Studien-/Prüfungsleistungen: | Studienleistungen: Hausarbeit, Präsentation, Projektarbeit Prüfungsleistungen: Klausur oder mündliche. Prüfung |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Soft Computing |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Embedded Intelligence, Anwendungsgebiet Embedded Intelligence, Basis Anwendungsgebiet Knowledge & Data Engineering, Anwendungsgebiet Knowledge & Data Engineering Anwendungsgebiet Robotik, |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 3. Semester |
| Dozent(in): | Prof. Dr. Bernhard Sick |
| Sprache: | Deutsch, Englisch nach Absprache |
| Zuordnung zum Curriculum | Informatik (Bachelor) |
| Lehrform/SWS: | 3 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz 120 Stunden Selbststudium, |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker |
| Angestrebte Lernergebnisse | Kenntnisse: wesentlichste Paradigmen aus dem Bereich des Soft Computing Fertigkeiten: praktischer Einsatz der Paradigmen (geübt unter Verwendung von Matlab) Kompetenzen: Bewertung von praktischen Anwendungen der Paradigmen, selbständige Entwicklung von einfachen Anwendungen. |
| Inhalt: | Die Vorlesung beschäftigt sich mit Neuronalen Netzen, Fuzzy-Logik und Evolutionären Algorithmen. Dieses Gebiet wird üblicherweise als "Soft-Computing" bezeichnet. Folgende Themen werden besprochen: Biologische Grundlagen, Überwacht lernende Neuronale Netze (z.B. Perzeptren, Mehrlagige Perzeptren, Radiale Basisfunktionen-Netze), Unüberwacht lernende Neuronale Netze (z.B. Wettbewerbslernen, Selbstorganisierende Karten), First-Order-Lernverfahren, Fuzzy-Logik und Fuzzy-Systeme, Genetische Algorithmen und Evolutionäre Verfahren, Anwendungsbeispiele (jeweils), Kombinationen verschiedener Verfahren |
| Studien-/Prüfungsleistungen: | Schriftliche Prüfung (120 min.) oder mündliche Prüfung (ca. 20 min.) |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Softwareergonomie |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik Anwendungsgebiet Software Tools |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Dozent(in): | Pfitzmann, Athanassiou |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Informatik (Master), Maschinenbau (Master) |
| Lehrform/SWS: | 1 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 30 Stunden Präsenz 60 Stunden Selbststudium, |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Programmierkenntnisse, Grundlagen der Informatik, Digitale Logik, Mikroprozessoren oder Rechnerarchitektur, Mathematik. |
| Angestrebte Lernergebnisse | <p>Angestrebte Lernergebnisse: Ziel der Veranstaltung ist es, Strategien zur Aneignung softwareergonomischen Wissens theoretisch und praktisch aufzuarbeiten und kritisch zu diskutieren. Der entscheidende Faktor für den optimalen Umgang mit technischen Systemen besteht dabei in der optimalen Gestaltung der Schnittstelle zwischen Mensch und Softwaresystem und dem dazugehörigen Wissen. Die Studierenden werden mit Themen der Softwareergonomie vertraut gemacht und lernen, sich mit neuesten Ergebnissen der Softwareergonomie-Forschung auseinanderzusetzen, sie zu vergleichen und kritisch zu evaluieren. Dabei werden die Teilnehmer in die Lage versetzt, die wichtigsten Theorien, Prinzipien und Methoden der Softwareergonomie beispielhaft einzusetzen und verschiedene Einsatzmöglichkeiten zu identifizieren. Die Studierenden sind in der Lage, die Erkenntnisse eigenständig auf andere Anwendungsbereiche der Softwaregestaltung zu übertragen, bzw. sie können herauszuarbeiten, welche Maßnahmen bei dem Einsatz softwareergonomischer Maßnahmen notwendig sind.</p> |
| Inhalt: | Zunächst wird erläutert, welche Funktion Modelle in der Mensch-Rechner-Interaktion haben können, welche Modellarten existieren und wofür sie eingesetzt werden. Anschließend werden verschiedene deskriptive Modelle vorgestellt, die als Leitbilder und Veranschaulichungshilfen für Softwareentwickler dienen können (z.B. das IFIP-Modell oder das Kommunikationsmodell von Oberquelle). Den Schwerpunkt in der Vorlesung bilden analytische Modelle, die vorhersagen, welches Interaktionswissen Benutzer |

| | |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>erwerben müssen (Kompetenzmodelle) und wie schnell Aufgaben mit Hilfe eines Systems gelöst werden können (Leistungsmodelle). Prototypische Modelle dieser Art sind die Task Action Grammar (TAG), das GOMS-Modell (goals, operators, methods, selection rules) und die Cognitive Complexity Theory (CCT). Thematische Schwerpunkte sind:</p> <ul style="list-style-type: none"> ▪ Einführung Softwareergonomie (Begriffe und Definitionen) ▪ Kognitive Analyse ▪ Modelle und Systeme ▪ Bilanzierung der SE ▪ Wissensidentifikation und Wissensbereitstellung ▪ Transferleistung und Informationsaustausch ▪ Gebrauchstauglichkeit ▪ Fallstudien |
| Studien-/Prüfungsleistungen: | Klausur (90 min.) oder mündliche Prüfung (30 min.) |

| | |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Softwarequalität |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Sem. |
| Modulverantwortliche(r): | Prof. Dr.-Ing. habil. Josef Börcsök |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mechatronik (Bachelor), Maschinenbau (Bachelor) |
| Lehrform/SWS: | Vorlesung /2 SWS ; Übung /2 SWS |
| Arbeitsaufwand: | Präsenzzeit: 2 SWS Vorlesung (30 Stunden); 2 SWS Übung (30 Std.) Selbststudium: 120 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Einführung in C, Mathematik-Veranstaltungen des Pflichtbereichs |
| Angestrebte Lernergebnisse | Erarbeitung der Merkmale von Softwarequalität, Zuverlässigkeit, Softwarezuverlässigkeitsmodellierung. Aufstellung und Bestimmung von Qualitätsmaßen für Software. |
| Inhalt: | Vorstellung und Wertung der Qualitätsmerkmale für Software. Kennenlernen und Analyse von Verfahren zur Qualitätssicherung Anwendung von mathematische Modellbeschreibungen. Beispielanwendungen aus verschiedenen Applikationen |
| Studien-/Prüfungsleistungen: | Klausur, mündliche. Prüfung, Hausarbeit, Referat/Präsentation |

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | SPS Programmierung nach IEC 61131-3 |
| ggf. Modulniveau | Wahlpflicht Technische Informatik |
| ggf. Kürzel | |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr. Michael H. Schwarz |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Maschinenbau (Bachelor), Mechatronik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 2 SWS Übung |
| Arbeitsaufwand: | 60 Stunden Präsenz, 120 Stunden Selbststudium |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Abgeschlossenes Grundstudium |
| Angestrebte Lernergebnisse | Die Studierenden lernen den Aufbau von Programmiersprachen nach IEC61131-3, entwickeln eine Methodenkompetenz zur Auswahl eines geeigneten Werkzeugs in Abhängigkeit vom Anwendungsbereich. |
| Inhalt: | Programmierung und Werkzeugauswahl, Vorstellung marktüblicher Werkzeuge mit Bezug auf die Anwendung, Beispielanwendungen |
| Studien-/Prüfungsleistungen: | Praktikumsbericht,/mdl. Prüfung/schriftl. Prüfung |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Techniken und Dienste des Internets |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Kommunikation und Verteilte Systeme, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | TDI |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortlicher: | Prof. Dr. Geihs |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Bachelor Informatik, Wahlpflicht |
| Lehrform / SWS: | 2 SWS Vorlesung und 2 SWS Übungen |
| Arbeitsaufwand: | 60 Stunden Präsenz 80 Stunden Übungen 20 Stunden Vor- und Nachbereitung von Vorlesung und Übung <u>20 Stunden Prüfungsvorbereitung</u> Summe: 180 Stunden |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundkenntnisse der Kommunikations- und Rechnertechnik |
| Angestrebte Lernergebnisse: | Praxisorientiertes Verständnis der Hilfsmittel zur Erstellung verteilter Anwendungen im Umfeld des Internet |
| Inhalt: | Die Vorlesung erläutert anwendungsnahe Protokolle, Dienste und Beschreibungsverfahren für die Erstellung von Internet-Anwendungen. Zu den Themen gehören: Internet-Architektur, Funktionsprinzipien der Protokolle, Datenbeschreibungssprachen, Anwendungsunterstützung, mobiler Code im Internet, Web Services (SOAP, WSDL; UDDI), Semantic Web, Sicherheit, Web 2.0 |
| Studien-/Prüfungsleistungen: | Klausur |

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Technik- und Produktrecht |
| ggf. Modulniveau | Basis Anwendungsgebiet Sozio-technisches Systemdesign, Anwendungsgebiet Sozio-technisches Systemdesign |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortliche(r): | Prof. Dr. Alexander Roßnagel |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor) und andere |
| Lehrform/SWS: | 2 SWS Vorlesung |
| Arbeitsaufwand: | Präsenzzeit: 30 Stunden Selbststudium: 60 Stunden |
| Kreditpunkte: | 3 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Angestrebte Lernergebnisse | Die Studierenden kennen die wichtigsten geltenden Vorschriften des Umweltrechts und des Wirtschaftsverwaltungsrechts und sie kennen das systematische Zusammenspiel auf unterschiedlichen Rechtsebenen. Sie kennen die ökologischen, politischen, wirtschaftlichen und technischen Grundlagen der rechtlichen Regelungen. Sie sind in der Lage, die erworbenen Kenntnisse zur Lösung von Fällen einzusetzen und problemangemessene Lösungen zu erarbeiten. |
| Inhalt: | Geräte- und Produktsicherheitsrecht, Produkthaftungsrecht, Rückruf von Produkten, öffentlich-rechtliche Fragestellungen betreffs die Hersteller von Produkten, Arbeitssicherheitsrecht, Compliance im Unternehmen |
| Studien-/Prüfungsleistungen: | Klausur oder Referat |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Technische Systeme im Zustandsraum |
| ggf. Modulniveau | Basis Anwendungsgebiet Modellierung und Simulation |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortlicher: | Prof. Dr. Linnemann |
| Sprache: | deutsch |
| Zuordnung zum Curriculum: | Informatik Bachelor, Elektrotechnik Bachelor |
| Lehrform / SWS: | 2 SWS Vorlesung und 1 SWS Übungen |
| Arbeitsaufwand: | 120 h: 45 h Präsenz, 75 h Selbststudium |
| Kreditpunkte: | 4 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Kenntnisse entsprechend der Inhalte der Module „Lineare Algebra“, „Analysis für Informatiker“ und „Elektrotechnik für Informatiker“. |
| Angestrebte Lernergebnisse: | Der/die Lernende kann <ul style="list-style-type: none"> - allgemeine lineare Netzwerke im Zustandsraum darstellen, - die Bedeutung von Differentialgleichungen erfassen, - die Lösung linearer Differentialgleichungen berechnen, - Methoden zur Lösung nichtlinearer Anfangswertaufgaben anwenden, - Simulationssoftware nutzen und zugrundeliegende Algorithmen skizzieren, - berechnete Lösungen interpretieren, - die Differentialgleichung einfacher technischer Systeme ermitteln. |
| Inhalt: | <ul style="list-style-type: none"> - Beschreibung linearer und nichtlinearer elektrischer Netzwerke durch Differentialgleichungen im Zustandsraum - Lösung linearer Differentialgleichungen im Zustandsraum - Lösung nichtlinearer Anfangswertaufgaben: Existenz und Eindeutigkeit, analytische Ansätze sowie numerische Verfahren - Beschreibung technischer Systeme durch Differentialgleichungen, Beispiele aus der Kinetik, Thermodynamik und Wellenausbreitung - Klassifikation von Differentialgleichungen: gewöhnlich, partiell, differentiell-algebraisch, Randwerte, etc. - Simulations- und Modellierungssoftware - Zeitdiskrete Systeme, Differenzgleichungen - Stabilität, Attraktoren |
| Studien-/Prüfungsleistungen: | Prüfungsleistung: Klausur (60 min.) oder mündliche Prüfung (30 min.) Studienleistung: Übungsaufgaben |

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Umweltwissenschaftliche Grundlagen für Ingenieure |
| ggf. Modulniveau | Basis Anwendungsgebiet Umweltingformatik |
| ggf. Kürzel | |
| Studiensemester: | Ab 3. Semester |
| Modulverantwortlicher: | Prof. Dr. Joseph Alcamo PD Dr. Rüdiger Schaldach Dr. Martina Flörke |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Informatik (Bachelor) und andere |
| Lehrform / SWS: | 2 SWS Vorlesung |
| Arbeitsaufwand: | 90 h: 30 h Präsenzzeit 60 h Selbststudium |
| Kreditpunkte: | 3 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | |
| Angestrebte Lernergebnisse: | Verständnis der grundlegenden Prinzipien der Umweltwissenschaften. |
| Inhalt: | Ziel der Lehrveranstaltung ist die Vermittlung von Kenntnissen über die grundlegenden Prinzipien der Umweltwissenschaften. Es werden insbesondere die Umweltbereiche Wasser, Klima, Luftverschmutzung sowie terrestrische Systeme behandelt. Dabei liegt der Schwerpunkt auf einer integrativen Betrachtung von naturwissenschaftlichen Aspekten und der anthropogenen Beeinflussung von Umweltgütern. Es wird ein systemorientierter Ansatz verfolgt, der auf dem Driver-Pressure-State-Impact-Response (DPSIR) Schema basiert. |
| Studien-/Prüfungsleistungen: | Seminarvortrag oder Klausur |

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Verteilte Systeme – Architekturen und Dienste |
| ggf. Modulniveau | Wahlpflicht Praktische Informatik, Basis Anwendungsgebiet Kommunikation und Verteilte Systeme, Anwendungsgebiet Kommunikation und Verteilte Systeme |
| ggf. Kürzel | VSAD |
| Lehrveranstaltungen: | |
| Studiensemester: | Ab 4. Semester |
| Modulverantwortlicher: | Prof. Dr. K. Geihs |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Wahlpflicht |
| Lehrform / SWS: | 2 SWS Vorlesung und 2 SWS Übungen |
| Arbeitsaufwand: | Präsenz 60 Übungen 80 Vor- und Nachbereitung von Vorlesung und Übung 20 <u>Prüfungsvorbereitung 20</u> Summe 180 |
| Kreditpunkte: | 6 CP |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Grundkenntnisse in Betriebssystemen und Kommunikationstechnik |
| Angestrebte Lernergebnisse: | Kenntnis und kritische Beurteilung der systemtechnischen Grundlagen und Alternativen von Verteilungsplattformen, praktischer Umgang mit Middleware-Produkten |
| Inhalt: | Die Vorlesung behandelt die systemtechnischen Grundlagen verteilter Systeme. Zu den Themen gehören Architekturen, Programmiermodelle, Dienste und grundlegende Mechanismen für Middleware-Plattformen, u. a. Client/Server, Message Queueing, Publish/Subscribe, RPC, CORBA, RMI, Jini, Enterprise Java Beans (EJB), .NET, Infrastrukturdienste (Verzeichnisse, Sicherheit, etc.) |
| Studien-/Prüfungsleistungen: | Bearbeitung von Übungsaufgaben und Klausur |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | Vertiefungsvorlesungen Mathematik (Mathematik: B9) |
| ggf. Modulniveau | Anwendungsgebiet Computational Mathematics |
| ggf. Lehrveranstaltungen | |
| Studiensemester: | Ab 5. Semester |
| Dozent(in): | Dozenten des FB 10 Mathematik und Naturwissenschaften |
| Sprache: | Meist Deutsch |
| Zuordnung zum Curriculum | Informatik (Bachelor), Mathematik (Bachelor) |
| Lehrform/SWS: | 2 SWS Vorlesung, 1 SWS Übung |
| Arbeitsaufwand: | 45 Stunden Präsenz, 135 Stunden Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Lineare Algebra, Analysis für Informatiker, Grundlagen der Algebra und Computeralgebra |
| Angestrebte Lernergebnisse | Mit diesem Modul sollen vertiefte Kenntnisse in einem Gebiet der Mathematik erworben werden, in dem die Bachelorarbeit geschrieben wird. |
| Inhalt: | Mögliche Themen der Vertiefungsvorlesungen sind z.B.: Computeralgebra II, Galoistheorie II, Kryptographie II, Elementare und algorithmische Zahlentheorie, Gröbnerbasen, Kodierungstheorie II, Markovketten und algorithmische Anwendungen, Stochastische Simulation, Schließende Statistik, Explorative Datenanalyse, Numerik gewöhnlicher Differentialgleichungen, Numerik linearer Gleichungssysteme, Eigenwertprobleme, Optimierung, Numerische lineare Algebra, Dynamische Systeme, Funktionenräume, Potentialtheorie, Integralgleichungen, Hilbertraummethode für partielle Differentialgleichungen, Halbgruppen, Wavelets, Spektraltheorie, Analysis auf Mannigfaltigkeiten, Differentialgeometrie, Einführung in die Mathematische Strömungsmechanik. Welche der jeweils aktuellen Lehrveranstaltungen diesem Modul zugeordnet sind, wird zusammen mit einer detaillierten Inhaltsbeschreibung im Vorlesungsverzeichnis des FB 10 (Modul B9) ausgewiesen. |
| Studien-/Prüfungsleistungen: | Studienleistung: Regelmäßige Bearbeitung von Übungsaufgaben; der Dozent kann für einzelne Lehrveranstaltungen die Bearbeitung der Übungsaufgaben auch ganz oder teilweise durch Kurzreferate oder Hausarbeiten ersetzen. Prüfungsleistung: Klausur oder mündliche Prüfung |

| | |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | VHDL – Kurs |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechner-technologie |
| Studiensemester: | Ab 6. Semester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | nach Absprache |
| Zuordnung zum Curriculum | Elektrotechnik (BSc.); Informatik (BSc.); Mechatronik (BSc.) |
| Lehrform/SWS: | 4 SWS: 2 SWS Vorlesung; 2 SWS Übung |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit 120 h Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | Digitale Logik |
| Angestrebte Lernergebnisse | Die/der Lernende – Grundelemente der Beschreibungssprache VHDL benennen – die Funktionsweise der Sprachelemente erläutern – in VHDL beschriebene Schaltungen interpretieren – Beschreibungen von Standardschaltungen in VHDL entwerfen, – mit Synthesesoftware Entwürfe implementieren. |
| Inhalt: | Syntax und Semantik von VHDL, verschiedene Modellierungsmöglichkeiten, Beschreibung von Standardfunktionalitäten (Schaltnetze, Zustandsautomaten, Datenpfadfunktionalität), Synthese von konkreten Schaltungen mit kommerzieller CAD-Software. |
| Klausur, Studienleistungen (b/bn): Übungsaufgaben | Klausur (90 Min.) oder mündl. Prüfung (etwa 40 Min.) |

| | |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulbezeichnung: | VHDL – Praktikum |
| ggf. Modulniveau | Wahlpflicht Technische Informatik, Anwendungsgebiet Prozessor- und Rechner-technologie |
| Studiensemester: | Ab 5. Semester |
| Modulverantwortliche(r): | Prof. Dr. Peter Zipf |
| Sprache: | nach Absprache |
| Zuordnung zum Curriculum | Elektrotechnik (BSc.); Informatik (BSc.); Mechatronik (BSc.) |
| Lehrform/SWS: | 4 SWS: Praktikum |
| Arbeitsaufwand: | 180 h: 60 h Präsenzzeit 120 h Selbststudium |
| Kreditpunkte: | 6 |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen: | VHDL-Kurs oder äquivalente LV, Digitale Logik, Rechnerarchitektur |
| Angestrebte Lernergebnisse | Die/der Lernende kann – ein komplexes Entwurfsprojekt planen und durchführen, – exemplarisch die Modellierung eines Prozessormodells mit Pipelining durchführen, – kommerzielle CAD-Programme zur Simulation und Validierung von Modellen anwenden, – kommerzielle CAD-Programme für Synthese und Charakterisierung von Modellen anwenden, – die Arbeiten eines Entwurfsteams organisieren. |
| Inhalt: | Entwurf einer komplexen Schaltung (z. \,B. eines Mikroprozessors) in kleinen Gruppen (z. \,B. je 4 Studenten); Aufgabenteilung innerhalb der Design-Teams; Systemmodellierung in VHDL; Simulation und Validierung der erstellten Modelle; Synthese auf ein FPGA, Charakterisierung der Implementierung; Test der Modelle auf einer Prototyp-Hardware. |
| Klausur, Studienleistungen (b/bn): Übungsaufgaben | Referat/Präsentation der Arbeiten, Bericht (Ausarbeitung, erstellter Code) und Teamarbeit bzw. Vorführung der Ergebnisse werden zu einer Gesamtmodulnote zusammengefasst |

Spezielle Module

| | |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Berufspraxis |
| Art des Moduls | Pflichtmodul |
| Anzahl Credits für das Modul | 12 |
| Angestrebte Lernergebnisse | Die Studierenden haben die berufliche und betriebliche Praxis in ein oder mehreren typischen Einsatzgebieten von Informatikern kennengelernt. |
| Lehrinhalte | |
| Lehr-/ Lernformen (Organisationsform) | Praktikum |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | 360 Stunden (i.d. Regel 9 Wochen) |
| Sprache | |
| Voraussetzungen laut Prüfungsordnung | soll frühestens nach Vorlesungszeit des vierten Fachsemesters absolviert werden |
| Studentischer Arbeitsaufwand | 360 Stunden |
| Studien- und Prüfungsleistung | Unbenoteter Bericht, nach Absprache mit Betreuer mündlich oder schriftlich (ca. 10 Seiten). |

| | |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modulname | Projekt |
| Art des Moduls | Wahlpflicht-Modul |
| Anzahl Credits für das Modul | 12 |
| Modulverantwortliche/r | Fohry |
| Angestrebte Lernergebnisse | Die Studierenden haben ihre Schlüsselkompetenzen Teamfähigkeit und Kommunikationsfähigkeit ausgebaut. Sie verfügen über vertiefte Kenntnisse in einem selbst gewählten Schwerpunktgebiet (entweder aus der Informatik oder aus einem Anwendungsgebiet). Weiterhin haben sie Erfahrung bei der eigenständigen Durchführung eines Projektes im Team gesammelt und ihre Fähigkeit zum selbständigen wissenschaftlichen Arbeiten hinführend auf die Bachelorarbeit vertieft. |
| Lehrinhalte | je nach Projekt |
| Lehr-/ Lernformen (Organisationsform) | Projektarbeit. Wahlweise können ein Projekt mit 12 CP oder zwei Projekte mit je 6 CP im gleichen oder unterschiedlichen Fachgebieten durchgeführt werden. |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | ein oder zwei Semester, Angebote in jedem Semester |
| Sprache | deutsch oder englisch |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen | Ab 5. Semester, weitere Voraussetzungen abhängig vom gewählten Gebiet |
| Studentischer Arbeitsaufwand | 360 Stunden, z.T. Präsenz |
| Studien- und Prüfungsleistung | Projektarbeit. Bei Modulteilprüfungsleistungen berechnet sich die Gesamtnote als mit CP gewichtetes arithmetisches Mittel. Studentisches Engagement kann mit max. 3 CP (unbenotet) eingebracht werden; in diesem Fall ergibt sich die Gesamtnote aus der/den übrigen Modulteilprüfungsleistung(en) |

| Modulname | Schlüsselkompetenzen |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Art des Moduls | Pflicht-/Wahlpflichtmodul: Die Veranstaltung Projektmanagement ist verpflichtend zu belegen. Zusätzlich sind Wahlpflichtveranstaltungen aus den Bereichen Betriebswirtschaftslehre, Recht, Managementtechnik, Fremdsprachen, Techniken wissenschaftlichen Arbeitens bzw. "Studentisches Engagement" zu belegen, wobei mindestens zwei der sechs Bereiche vertreten sein sollen. Unter "Studentisches Engagement" fallen Tutorentätigkeiten und die Mitarbeit in der studentischen Selbstverwaltung (Richtlinien des Prüfungsausschusses beachten). |
| Anzahl Credits für das Modul | 9 |
| Angestrebte Lernergebnisse | Die Studierenden verfügen über Grundkenntnisse in den gewählten Bereichen sowie über Kompetenzen in Projektmanagement und fachübergreifendem Lernen. |
| Lehrinhalte | Kenntnisse bzw. Kompetenzen in o.g. Bereichen Die Veranstaltung Projektmanagement vermittelt fachübergreifend die Grundelemente des Projektmanagements und zeigt den Studierenden Bedeutung und Wert des PM im Arbeitsleben und bei der Bewältigung von Fachaufgaben. |
| Lehr-/ Lernformen (Organisationsform) | Bei PM: Vorlesung, Übung. Darüber hinaus abhängig von den gewählten Veranstaltungen |
| Verwendbarkeit des Moduls | Informatik Bachelor und weitere Studiengänge |
| Dauer und Häufigkeit des Angebotes | Dauer abhängig von gewählten Veranstaltungen, Angebote in jedem Semester. Liste der wählbaren Veranstaltungen wird auf der Webseite des Studiengangs veröffentlicht. |
| Sprache | deutsch oder englisch |
| Studentischer Arbeitsaufwand | 270 Stunden, davon ca. 90 Stunden Präsenz |
| Studien- und Prüfungsleistung | Prüfungsleistung: Vortrag (30 – 45 min.), Klausur (60 – 180 min.), mündliche Prüfung (20 – 40 min.), Hausarbeit (15 – 20 Seiten), Projektarbeit oder Abschlussbericht. Je nach gewählten Veranstaltungen sind evtl. Studienleistungen erforderlich. Form der Studien-/Prüfungsleistungen wird jeweils zu Veranstaltungsbeginn bekanntgegeben. Bei Modulteilprüfungsleistungen berechnet sich die Gesamtnote als mit CP gewichtetes arithmetisches Mittel. Studentisches Engagement kann mit max. 3 CP (unbenotet) eingebracht werden; in diesem Fall ergibt sich die Gesamtnote aus der/den übrigen Modulteilprüfungsleistung(en) |

| Modulname | Seminar |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Art des Moduls | Wahlpflichtmodul |
| Anzahl Credits für das Modul | 4 |
| Modulverantwortliche/r | Fohry |
| Angestrebte Lernergebnisse | Die Studierenden haben ihre Schlüsselkompetenzen in den Bereichen Literaturarbeit und Darstellungstechniken hinführend auf die Bachelorarbeit ausgebaut. Sie verfügen über vertiefte Kenntnisse in einem selbst gewählten Schwerpunktgebiet (entweder aus der Informatik oder aus einem Anwendungsgebiet). |
| Lehrinhalte | je nach Seminar |
| Lehr-/ Lernformen (Organisationsform) | Seminar 2 SWS |
| Verwendbarkeit des Moduls | Informatik Bachelor |
| Dauer und Häufigkeit des Angebotes | ein Semester, Angebote in jedem Semester |
| Sprache | deutsch oder englisch |
| Voraussetzungen nach Prüfungsordnung | Mathematiktest |
| Empfohlene Voraussetzungen | Grundstudium, weitere Voraussetzungen abhängig vom gewählten Gebiet |
| Studentischer Arbeitsaufwand | 120 Stunden, davon 30 Stunden Präsenz |
| Studien- und Prüfungsleistung | Vortrag (30 - 45 min.) und Hausarbeit (15 - 20 Seiten) |
| Anzahl Credits für das Modul | 4 |