

**Master Studiengang
Angewandte Informatik**

PO 20

Modulhandbuch

Erstellt am: 6. Oktober 2021

Wo finde ich Hilfe während des Studium?

Wichtige Informationsquellen rund um das Studium der Angewandten Informatik:

Studiengangsw Webseite:

<https://www.ai.ini.rub.de/>

Studienfachberatung Angewandte Informatik:

<https://ai.ini.rub.de/kontakte/studienberatung>

Prüfungsamt Angewandte Informatik:

<https://ai.ini.rub.de/kontakte/pruefungsamt>

Fachschaftsrat Angewandte Informatik:

<https://ai-rub.de/>

Bei fachlichen Fragen besteht die Möglichkeit die Dozenten während Ihrer Sprechstunden (siehe individuelle Webseiten) zu kontaktieren.

Weitere wichtige Kontaktadressen auf dem Campus sind:

Zentrale Studienberatung:

<https://www.ruhr-uni-bochum.de/zsb/>

Bietet Hilfe und Coaching bei individuellen Problemen (auch psychologische Betreuung).

Studienfinanzierungsberatung:

<https://studium.ruhr-uni-bochum.de/de/studienfinanzierung>

Talentscouts der RUB:

<https://studium.ruhr-uni-bochum.de/de/die-talentscouts-der-rub>

Beratung zu Stipendien; hierzu werden extra Workshops angeboten.

Beratungszentrum zur Inklusion Behinderter:

<https://www.akafoe.de/inklusion/>

International Office:

<http://www.international.rub.de/ausland/index.html.de>

Beratung zu Studienaufenthalten im Ausland

Wohnheimplätze:

<https://www.akafoe.de/wohnen/>

Studiengangsziele:

Im Masterstudiengang werden gemäß DQR Niveaustufe 7 Kompetenzen im Bereich der Angewandten Informatik vermittelt, die zur Bearbeitung von neuen komplexen Aufgaben- und Problemstellungen sowie zur eigenverantwortlichen Steuerung von Prozessen benötigt werden. Absolventinnen und Absolventen verfügen über umfassendes, detailliertes und spezialisiertes Wissen auf dem neuesten Erkenntnisstand in verschiedenen Anwendungsbereichen der Informatik. Sie nutzen spezialisierte fachliche und konzeptionelle Fertigkeiten zur Lösung von Problemen und können darüber hinaus auch neue Verfahren unter Berücksichtigung gegebener Rahmenbedingungen entwickeln. Fehlendes Wissen erschließen Sie sich selbstständig. In Gruppenprojekten tragen Sie verantwortlich zur Lösung komplexer Aufgabenstellungen bei und vertreten die Ergebnisse vor anderen. Darüber hinaus sind Absolventinnen und Absolventen des Masterstudienganges in der Lage fachspezifische Diskussionen auch in englischer Sprache zu führen. Der Umgang mit englischer Fachliteratur ist eine Selbstverständlichkeit. Je nach Wahl der freien Wahlmodule können noch weitere fachübergreifende Kompetenzen erlangt werden.

Modularisierungskonzept:

Das Studium ist modular aufgebaut. Die Module stellen zeitlich und inhaltlich abgeschlossene Teilqualifikationen dar. Sie haben, abgesehen von wenigen Ausnahmen, einen Workload von mindestens 5 Creditpoints (CP), wobei ein Creditpoint in etwa einer Arbeitsbelastung von 30 Stunden entspricht. In die Berechnung des Workloads fließt neben der Präsenzzeit auch die Zeit für das Selbststudium mit ein (Bearbeitung von Übungsaufgaben, Vor- und Nachbereitung der Vorlesungen, Lesen geeigneter Literatur, ...). Im Masterstudium gibt es keine Pflichtmodule. Die im Wahlpflichtbereich angesiedelten Module erweitern das Grundlagenwissen aus dem Bachelorstudium, um den spezialisierteren Vertiefungsmodulen folgen zu können. Insgesamt bietet das Studium zahlreiche Wahlmöglichkeiten für die Studierenden, um ein individuelles Studienprofil zu entwickeln.

Prüfungsformen:

Prüfungsleistungen können in Form einer Klausur (auch in elektronischer Form), einer mündlichen Prüfung, eines Seminarbeitrags, eines Referates oder Präsentation, einer Hausarbeit, einer Projektarbeit, einer praktischen Prüfung oder eines Kolloquiumsvortrags erbracht werden.

Studienplan Master Angewandte Informatik PO 20

| Nr | Modul | Lehrveranstaltung | Umfang bzw. Mind. Umfang (CP) | Empfohlenes Semester | Bewertung |
|---------------------------|-------------------------------------|-----------------------------|---|----------------------|-----------|
| Wahlpflichtbereich | | | | | |
| 1 | Wahlpflichtmodule* | Wahlpflichtmodule ** | 20* | 1-3 | benotet |
| Anwendungsbereich | | | | | |
| 2 | Anwendungsmodule | Anwendungsmodule** | 35* | 1-3 | benotet |
| 3 | Fachwissenschaftliche Vertiefung | Vertiefungsseminare** | 6 | 1-3 | benotet |
| Freier Wahlbereich | | | | | |
| 4 | Freie Wahlmodule | Freie Wahlmodule*** | 15 | 1-3 | unbenotet |
| Wahlpflichtbereich | | | | | |
| 5 | Master-Studienprojekt | Master-Studienprojekt | 10 | 3 | benotet |
| Masterarbeit | | | | | |
| 6 | Masterarbeit und Kolloquium | Masterarbeit und Kolloquium | 30 | 4 | benotet |
| Summe: | | | 120 | | |

* Der Gesamtumfang der Wahlpflichtmodule und Anwendungsmodule darf 59 CP nicht unterschreiten.

** Informationen zu den wählbaren Wahlpflicht- und Anwendungsmodulen befinden sich im jeweils aktuellen Modulhandbuch. Informationen zu den angebotenen Seminaren finden Sie im Vorlesungsverzeichnis der RUB.

*** Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der Universitätsallianz Ruhr gewählt werden.

| Lehrveranstaltung | Einheit | Umfang Modul (LP) | Semester | Bewertung |
|--|------------|----------------------|--------------|-----------|
| Wahlpflichtmodule | | | | |
| Effiziente Algorithmen | Mathe | 9 | SS | benotet |
| Geometrische Algorithmen | Mathe | 9 bzw. 6 im WS 21/22 | WS | benotet |
| Fundamentals of GPU Programming | ETIT | 5 | WS | benotet |
| Parallel Computing | Baulng | 6 | SS | benotet |
| Machine Learning: Evolutionary Algorithms | INI | 6 | WS | benotet |
| Machine Learning: Supervised Methods | INI | 6 | SS | benotet |
| Machine Learning: Unsupervised Methods | INI | 6 | WS | benotet |
| Kryptographie | Mathe | 8 | WS | benotet |
| Komplexitätstheorie (erst wieder SS 22) | Mathe | 9 | unregelmäßig | benotet |
| Anwendungsmodule | | | | |
| Ingenieurinformatik | | | | |
| Design Optimization | Baulng | 6 | WS | benotet |
| Design soziotechnischer Informationssysteme | AW | 5 | WS | benotet |
| IT im Engineering | MB | 6 | WS | benotet |
| Verkehrstechnik | Bauing | 6 | SS | benotet |
| Grundlagen der automatischen Spracherkennung | ETIT | 6 | SS | benotet |
| Programmier- und Simulationstechnik | | | | |
| Simulationstechnik | Baulng | 5 | WS | benotet |
| 3D-Simulation in der Automatisierungstechnik | MB | 6 | WS | benotet |
| Complexity Economics and Agent-based Modeling | WiWi | 10 | WS | benotet |
| High-Performance Computing on Clusters | Bauing | 6 | WS | benotet |
| High-Performance Computing on Multi- and Manycore Processors | Bauing | 6 | SS | benotet |
| Knowledge Graphs | Mathe | 5 | SS | benotet |
| Neuroinformatik | | | | |
| Autonomous Robotics: Action, Perception and Cognition | INI | 6 | SS | benotet |
| Computational Neuroscience: Neural Dynamics | INI | 6 | WS | benotet |
| Computational Neuroscience: Vision and Memory | INI | 5 | SS | benotet |
| Master-Praktikum: Autonomous Robotics | INI | 3 | WS/SS | benotet |
| Deep Learning | Mathe | 5 | WS | benotet |
| Kryptologie und theoretische Informatik | | | | |
| Kryptanalyse 1 | Mathe | 5 | SS | benotet |
| Symmetrische Kryptanalyse | Mathe | 5 | WS | benotet |
| Kryptographische Protokolle | Mathe | 5 | SS | benotet |
| Quantenalgorithmen (nicht im SS 21) | Mathe | 5 | SS | benotet |
| Systemsicherheit | ETIT | 5 | SS | benotet |
| Model Checking (nicht im WS 22/23, ab SS 23 jeweils im SS) | Mathe | 5 | WS | benotet |
| IT im Management | | | | |
| Groupware und Wissensmanagement | AW | 5 | SS | benotet |
| Industrielles Kundenmanagement | MB | 5 | SS | benotet |
| Product Lifecycle Management | MB | 6 | SS | benotet |
| Managing Digital Platform Ecosystems | TUDo | 7 | SS | benotet |
| Computerlinguistik | | | | |
| Anwendungen der Computerlinguistik (ab SS 22) | Philologie | 10 bzw. 12 | SS/WS | benotet |
| Bioinformatik | | | | |
| Bioinformatik in der Proteomik I | MPC | 5 | WS | benotet |
| Bioinformatik in der Proteomik II | MPC | 5 | SS | benotet |
| Bioimage Informatics | Biologie | 5 | WS | benotet |
| Master-Praktikum: Big Data in der Bioinformatik | Biologie | 5 | WS/SS | benotet |
| Master-Praktikum: Computational Proteomics | MPC | 5 | SS | benotet |

Abkürzungen:

AW: Institut für Arbeitswissenschaften
Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften
ETIT: Fakultät für Elektrotechnik und Informationstechnik
INI: Institut für Neuroinformatik
MB: Fakultät für Maschinenbau
MPC: Medizinisches Proteom Center
WIWI: Fakultät für Wirtschaftswissenschaft
TUDo: Technische Universität Dortmund

SS: Sommersemester
WS: Wintersemester

CP: Creditpoints

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Wahlpflichtmodule | 1 |
| 1.1 | Effiziente Algorithmen | 1 |
| 1.2 | Geometrische Algorithmen | 3 |
| 1.3 | Fundamentals of GPU Programming | 5 |
| 1.4 | Parallel Computing | 6 |
| 1.5 | Machine Learning: Evolutionary Algorithms | 8 |
| 1.6 | Machine Learning: Supervised Methods | 10 |
| 1.7 | Machine Learning: Unsupervised Methods | 12 |
| 1.8 | Kryptographie | 14 |
| 1.9 | Komplexitätstheorie | 16 |
| 2 | Anwendungsbereich | 18 |
| 2.1 | Design Optimization | 18 |
| 2.2 | Design sozio-technischer Informationssysteme | 20 |
| 2.3 | IT im Engineering | 22 |
| 2.4 | Verkehrstechnik | 24 |
| 2.5 | Grundlagen der automatischen Spracherkennung | 26 |
| 2.6 | Simulationstechnik | 28 |
| 2.7 | 3D-Simulation in der Automatisierungstechnik | 30 |
| 2.8 | Complexity Economics and Agent-based Modeling | 32 |
| 2.9 | High Performance Computing on Clusters | 34 |
| 2.10 | High-Performance Computing on Multi- and Manycore Processors | 36 |
| 2.11 | Knowledge Graphs | 38 |
| 2.12 | Autonomous Robotics: Action, Perception and Cognition | 40 |
| 2.13 | Computational Neuroscience: Neural Dynamics | 42 |
| 2.14 | Computational Neuroscience: Vision and Memory | 44 |
| 2.15 | Master-Praktikum: Autonomous Robotics | 46 |
| 2.16 | Deep Learning | 48 |
| 2.17 | Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse) | 49 |
| 2.18 | Symmetrische Kryptoanalyse | 51 |
| 2.19 | Kryptographische Protokolle | 52 |
| 2.20 | Quantenalgorithmen | 54 |
| 2.21 | Systemsicherheit | 56 |
| 2.22 | Model Checking | 58 |
| 2.23 | Groupware und Wissensmanagement | 60 |
| 2.24 | Industrielles Kundenmanagement | 62 |
| 2.25 | Product Lifecycle Management | 64 |
| 2.26 | Managing Digital Platform Ecosystems | 66 |

| | | |
|----------|--|-----------|
| 2.27 | Anwendungen der Computerlinguistik | 68 |
| 2.28 | Bioinformatics for Proteomics I | 70 |
| 2.29 | Bioinformatics for Proteomics II | 73 |
| 2.30 | Bioimage Informatics | 76 |
| 2.31 | Master-Praktikum Big Data in der Bioinformatik | 78 |
| 2.32 | Master-Praktikum Computational Proteomics | 81 |
| 2.33 | Fachwissenschaftliche Vertiefung | 83 |
| 3 | Freie Wahlmodule | 85 |
| 3.1 | Freie Wahlmodule | 85 |
| 4 | Studienprojekt | 87 |
| 4.1 | Master-Studienprojekt | 87 |
| 5 | Masterarbeit und Kolloquium | 89 |
| 5.1 | Masterarbeit | 89 |

Wahlpflichtmodule

| Effiziente Algorithmen | | | | | |
|---|-----------------|------------------|---|--------------------------------|----------------------------------|
| Modul-Nr: | Credits: | Workload: | Semester: | Turnus: | Dauer: |
| 1a | 9 CP | 270 h | 1.-3. Sem. | jedes SoSe | 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Effiziente Algorithmen (150320) b) Übung (150321) | | | Kontaktzeit: a) 4 SWS b) 2 SWS | Selbststudium: 180 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Empfohlene Vorkenntnisse: Die Inhalte der Veranstaltung "Datenstrukturen" bzw. "Informatik 2". | | | | | |
| Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls: Die Studierende <ul style="list-style-type: none"> • Kennen, auswählen und nutzen grundlegende Datenstrukturen und Graphenalgorithmen • Sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und beurteilen • Können auch bei praktischen Problemen entscheiden, welche der vermittelten Methoden/Algorithmen/Datenstrukturen anwendbar sind und diese nach Effizienz (insb. Laufzeit der Algorithmen) bewerten • Können konkrete Anwendungsprobleme modellieren und bei Bedarf diese Algorithmen weiter entwickeln | | | | | |
| Inhalt: Die Lehrveranstaltung kann sowohl in das Gebiet der praktischen als auch in das Gebiet der theoretischen Informatik eingeordnet werden. Die zentralen Themen sind die Folgenden: <ul style="list-style-type: none"> • Berechnung kürzester Pfade in Digraphen | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

- Berechnung eines maximalen Flusses in einem Transportnetzwerk
- Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt)

Darüberhinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme.

Lehrformen:

Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen

Prüfungsformen:

Abschlussprüfung; Klausur (120 Minuten)

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussprüfung

Verwendung des Moduls (in anderen Studiengängen):

B.Sc. Mathematik, M.Sc. Mathematik, M.Sc. IT-Sicherheit / Informationstechnik

Stellenwert der Note für die Endnote: 9 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

PD Dr. Daniela Kacso

| Geometrische Algorithmen | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--|-----------------------------|
| Modul-Nr: 1b | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: Jedes zweite Semester, idR im WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Geometrische Algorithmen (150341) b) Übung (150342) | | | Kontaktzeit: 6 SWS | Selbststudium: 90 h | Gruppengröße: 20 |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Grundlegende Kenntnisse ueber Algorithmen und Datenstrukturen. Empfohlene Vorkenntnisse: Grundlagen der Stochastik. | | | | | |
| Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende geometrische Algorithmen und Datenstrukturen • können Studierende Algorithmen nach dem Sweep-Paradigma analysieren und entwerfen • können Studierende inkrementelle Algorithmen entwerfen und analysieren, insbesondere randomisiert inkrementelle Algorithmen • können Studierende geometrische Algorithmen nach dem Teile-und-Herrsche Prinzip analysieren und entwerfen • können Studierende für Bereichsanfragen geeignete Datenstrukturen aussuchen | | | | | |
| Inhalt: Die Algorithmische Geometrie beschäftigt sich mit dem Entwurf und der Analyse von Algorithmen und Datenstrukturen für geometrische Probleme. Dazu werden zunächst einige grundlegende Probleme betrachtet, wie das Berechnen der konvexen Hülle einer Punktmenge, der Schnittpunkte einer Menge von Strecken oder einer Triangulierung eines einfachen Polygons. Anschließend sehen wir Algorithmen zum Berechnen bekannter geometrische Strukturen, wie das Voronoi-Diagramm, die Delaunay-Triangulierung und Arrangements. Ebenfalls betrachten wir Datenstrukturen für effiziente Anfragen auf geometrischen Daten, wie Rangetrees, kd-Bäume und Quadrees. Dabei kommen vor allem drei Arten von Algorithmen zum Einsatz: inkrementell, teile-und-herrsche, und sweep. Manche von diesen treten als randomisierte Algorithmen auf. | | | | | |
| Lehrformen: Vorlesung als kombinierter Folien- und Tafelvortrag, Übungen mit begleitendem Implementierungsprojekt | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

| |
|--|
| Prüfungsformen: Abschlussprüfung; Mündliche Prüfung |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene mündliche Prüfung, sowie als Studienleistung erfolgreiches Bearbeiten der Übungen und des Projektes mit Abgabe eines Projektberichtes |
| Verwendung des Moduls (in anderen Studiengängen): B.Sc. Mathematik, M.Sc. Mathematik |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Maike Buchin |
| Sonstige Informationen: Literatur: Die Vorlesung orientiert sich groesstenteils an dem Buch "Computational Geometry: Algorithms and Applications", von Mark de Berg, Otfried Cheong, Marc van Kreveld, und Mark Overmars (3te Auflage, 2008, Springer). |

KAPITEL 1. WAHLPFLICHTMODULE

| Fundamentals of GPU Programming | | | | | |
|---|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 1c | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Fundamentals of GPU Programming (141374) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: C (programming language) | | | | | |
| Lernziele (learning outcomes): To learn how to program on graphics processing units (GPUs). | | | | | |
| Inhalt: <ol style="list-style-type: none"> 1. GPU as a modern means for general-purpose massively parallel computations 2. General GPU architecture and CUDA operational model 3. Basic CUDA syntax 4. Optimization strategies in GPU programming 5. Case study of general-purpose GPU programming | | | | | |
| Lehrformen: Vorlesung | | | | | |
| Prüfungsformen: Abschlussprüfung; Mündliche Prüfung, 30 Minuten- | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene mündliche Prüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M. Sc. IT-Sicherheit/Informationstechnik | | | | | |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |
| Modulbeauftragte/r und hauptamtlich Lehrende: Dr. Denis Eremin | | | | | |

| Parallel Computing | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 1d | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Parallel Computing (127501) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Programmierkenntnisse in den Sprachen C, C++ oder Java | | | | | |
| <p>Lernziele (learning outcomes): Goals of the course “Parallel Computing” include a basic understanding of the theoretical foundation for efficient parallel algorithms, the architecture of parallel hardware systems and current parallel software paradigms used in research and industry today. Also, students learn how to develop parallel algorithms and implement them using state-of-the-art software systems and programming tools. In particular, numerically intensive engineering applications are a focus within the lecture itself as well as a target for the team projects carried out by the students of this course. Further, students must clearly present their projects results in a classroom setting to an audience with various technical background (the course is attended by students from applied computer science as well as by students from computational engineering).</p> <p>When the students have successfully complete this course,</p> <ul style="list-style-type: none"> • students will be familiar with the development of parallel algorithms and parallel software systems and have a good understanding of how to realize efficient parallel programs; • students will have a good understanding of the basics of parallel programming to be able to, for example, select a proper parallel software system or tool that is appropriate for a given numerical engineering task, or be able to discuss possible solutions concerning necessary hardware or software requirements at a competent level. | | | | | |
| <p>Inhalt:</p> <ul style="list-style-type: none"> • Introduction to parallel computing using examples from the field of engineering • Theoretical foundations of parallel computing (concurrency, parallel processes, deadlocks, Amdahl’s Law, Flynn’s taxonomy, efficiency metrics, memory models, etc.) • Parallel programming based on “shared memory” using the OpenMP application programming interface • Parallel programming based on “distributed memory” based the Message Passing Interface (MPI) | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

- Hardware based parallel programming based on the General Purpose Graphical Processing Unit (GPGPU)
- Application of parallel programming paradigms to solve engineering tasks as a team project

Lehrformen:

- lectures
- exercises using computers
- team projects

Prüfungsformen:

Semesterbegleitend; - completion of a team project (2-4 students) - presentation of project results - oral examination

Voraussetzungen für die Vergabe von Kreditpunkten:

- successful completion of a team project
- presentation of project results in the classroom
- oral examination

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Computational Engineering

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. M. König, Dr. K. Lehner

Sonstige Informationen:

Literatur:

1. A. Schill, Th. Springer, "Verteilte Systeme", Springer-Verlag, 2007
2. Th. Raubner, G. Rünger, "Parallele Programmierung", Springer-Verlag, 2007
3. G. Bengel, et. al., "Masterkurs Parallele und Verteilte Systeme", Vieweg+Teubner, 2008

| Machine Learning: Evolutionary Algorithms | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 1e | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Machine Learning: Evolutionary Algorithms (310008) b) Übung (310018) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Der Kurs ist für Studierende des Masterstudiengangs Angewandte Informatik ausgerichtet. Teilnehmer sollten mit Linearer Algebra und elementarer Wahrscheinlichkeitstheorie vertraut sein. | | | | | |
| Lernziele (learning outcomes): Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Teilnehmer die wichtigsten Klassen direkter Optimierungsverfahren und ihre algorithmischen Komponenten, • haben die Teilnehmer ein tiefes Verständnis evolutionärer Algorithmen, insbesondere für kontinuierliche Probleme, • kennen die Teilnehmer eine Reihe spezifischer Problemschwierigkeiten und die zugehörigen algorithmischen Komponenten, welche diese Adressieren, • können die Teilnehmer elementare Laufzeitanalysen durchführen und verstehen die wichtigsten Konvergenzklassen • können die Teilnehmer Optimierungsverfahren selbst implementieren und zur Lösung neuer Probleme anwenden. | | | | | |
| Inhalt: Breiter Überblick über Optimierungsverfahren. Evolutionäre Optimierungsverfahren für black-box Optimierungsverfahren Algorithmische Komponenten für schlechte Konditionierung, Multimodalität, Rauschen, Nebenbedingungen und Mehrzieloptimierung. Konvergenz- und Laufzeitanalyse. | | | | | |
| Lehrformen: flipped classroom | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

| |
|---|
| Prüfungsformen: Abschlussprüfung; Klausur 90 Minuten, Wiederholungsklausur nach 6 Monaten |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Jun.-Prof. Dr. Tobias Glasmachers |

| Machine Learning: Supervised Methods | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|---|
| Modul-Nr: 1f | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Machine Learning: Supervised Methods (310508) b) Übung (310518) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig, zuletzt 80 Teilnehmer |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Vorlesung: Mathematics for Modeling and Data Analysis | | | | | |
| <p>Lernziele (learning outcomes): Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verstehen die Teilnehmer die Grundlagen der statistischen Lerntheorie, • kennen die Teilnehmer die wichtigsten Algorithmen des überwachten statistischen Lernens und können diese auf Lernprobleme anwenden, • kennen die Teilnehmer Stärken und Beschränkungen verschiedenen Lernverfahren, • können die Teilnehmer Standardsoftware zum maschinellen Lernen zur Lösung neuer Probleme einsetzen. | | | | | |
| Inhalt: Grundlagen der statistischen Lerntheorie, Querschnitt der wichtigsten Algorithmen des maschinellen Lernens, konkrete Problemlösung mit Standardsoftware | | | | | |
| Lehrformen: flipped classroom | | | | | |
| Prüfungsformen: Abschlussprüfung; Klausur 90 Minuten, Wiederholungsklausur nach 6 Monaten | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |

Modulbeauftragte/r und hauptamtlich Lehrende:

Jun.-Prof. Dr. Tobias Glasmachers

Sonstige Informationen:

Es bietet sich an, diese Vorlesung in Kombination mit der Veranstaltung "Machine Learning: Unsupervised Methods", angeboten jeweils im Wintersemester, zu hören.

| Machine Learning: Unsupervised Methods | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|-----------------------------|
| Modul-Nr: 1g | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Machine Learning: Unsupervised Methods (310003) b) Übung (310013) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: 40 |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: The mathematical level of the course is mixed but generally high, including calculus (functions, derivatives, integrals, differential equations, ...), linear algebra (vectors, matrices, inner product, orthogonal vectors, basis systems, ...), and a bit of probability theory (probabilities, probability densities, Bayes' theorem, ...). Programming is done in Python, thus the students should have a basic knowledge of that as well, or at least be fluent in another programming language. | | | | | |
| Lernziele (learning outcomes): After the successful completion of this course the students: <ul style="list-style-type: none"> • know a number of important unsupervised learning methods, • can discuss and decide which of the methods are appropriate for a given data set, • understand the mathematics of these methods, • know how to implement and apply these methods in python, • have gained experience in organizing and working in a team, • know problem solving strategies like brain storming, • can communicate about all this in English. | | | | | |
| Inhalt: This course covers a variety of shallow unsupervised methods from machine learning such as principal component analysis, independent component analysis, vector quantization, clustering, Bayesian theory and graphical models. | | | | | |
| Lehrformen: This course is given in a hybrid of inverted classroom and problem based learning. The course starts with a two-week introduction into unsupervised methods of machine learning, providing an overview. The students then work in groups of about 4 on realistic problems that can be solved with these methods. In the first week of a problem, they develop hypotheses and strategies for a solution and identify which methods they want to learn. Then the course agrees on a method to focus on theoretically, which will then be done in an inverted classroom | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

format. The students then try to solve the problem and present their results in a short talk with slides. Thus the students will not only learn about machine learning but also soft skills.

Prüfungsformen:

Semesterbegleitend; The exam is a combination of graded presentations for the problems and graded quizzes for the theory. 40% of the grade come from the average group performance on solving the problems. 10% come from the presentations, taking into account slides and presentation style, this is an individual grade of the presenter. 50% come from a digital quiz about the theory of the methods covered. Thus 60% of the grade are individual, 40% come from the group. In addition you can gain up to 8 bonus points for (i) being voted for as a 'most valuable player (MVP)' on a project and (ii) creating useful quiz questions.

Voraussetzungen für die Vergabe von Kreditpunkten:

Continuous participation and passed exam.

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Elektrotechnik und Informationstechnik

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Laurenz Wiskott

Sonstige Informationen:

Sessions with teacher (formally tutorial and lecture): Takes place every week on Tuesday from 10:30 to 12:00 and from 12:15 to 13:45 in an online session.; First appointment is on 12.10.2021; Last appointment is on 01.02.2022

| Kryptographie | | | | | |
|--|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 1h | Credits: 8 CP | Workload: 240 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Kryptographie (150312) b) Übung (150313) | | | Kontaktzeit: 6 SWS | Selbststudium: 150 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Empfohlene Vorkenntnisse: Keine | | | | | |
| Lernziele (learning outcomes): Die Studierenden haben ein Verständnis der wesentlichen mathematischen Methoden und Verfahren, auf denen moderne kryptographische Verfahren beruhen. Die Tiefe der Behandlung der Verfahren geht deutlich über das in den vorhergehenden Veranstaltungen vermittelte Maß hinaus. Die Teilnehmer sind zur Analyse und dem Design aktueller und zukünftiger kryptographischer Methoden befähigt. Zudem weisen sie ein Bewusstsein für Methodik und Mächtigkeit verschiedenster Angriffsszenarien auf. | | | | | |
| Inhalt: Es wird eine Einführung in moderne Methoden der symmetrischen und asymmetrischen Kryptographie geboten. Dazu wird ein Angreifermodell definiert und die Sicherheit der vorgestellten Verschlüsselungs-, Hash- und Signaturverfahren unter wohldefinierten Komplexitätsannahmen in diesem Angreifermodell nachgewiesen. | | | | | |
| <ul style="list-style-type: none"> • Sichere Verschlüsselung gegenüber KPA-, CPA- und CCA-Angreifern • Pseudozufallsfunktionen und -permutationen • Message Authentication Codes • Kollisionsresistente Hashfunktionen • Blockchiffren • Konstruktion von Zufallszahlengeneratoren • Diffie-Hellman Schlüsselaustausch • Trapdoor Einwegpermutationen • Public Key Verschlüsselung: RSA, ElGamal, Goldwasser-Micali, Rabin, Paillier • Einwegsignaturen • Signaturen aus kollisionsresistenten Hashfunktionen | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

| |
|--|
| <ul style="list-style-type: none">• Random-Oracle Modell |
| Lehrformen: Vorlesung und Übung |
| Prüfungsformen: Abschlussprüfung; Klausur (120 Minuten) |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. IT-Sicherheit / Netze und Systeme, B.Sc. IT-Sicherheit / Informationstechnik |
| Stellenwert der Note für die Endnote: 8 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Alexander May |
| Sonstige Informationen: Diese Veranstaltung ist im Vorlesungsverzeichnis der Mathematik als "Kryptographie I + II" aufgeführt. Literatur: Katz, Lindell, "Introduction to Modern Cryptography", Chapman und Hall/CRC, 2008 |

| Komplexitätstheorie | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: li | Credits: 9 CP | Workload: 270 h | Semester: 1.-3. Sem. | Turnus: unregelmäßig | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Komplexitätstheorie (150262) b) Übung (150263) | | | Kontaktzeit: 6 SWS | Selbststudium: 180 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Vorlesung über Theoretische Informatik | | | | | |
| Lernziele (learning outcomes): Die Studierenden lernen, algorithmische Probleme bezüglich ihrer Komplexität einzuordnen und so geeignete algorithmische Techniken zu ihrer Lösung zu identifizieren. Sie können insbesondere algorithmische Methoden für NP-vollständige Probleme anwenden. Sie können mit unterschiedlichen Berechnungsmodellen umgehen und sind in der Lage, einfache Aussagen über sie zu beweisen. Sie lernen im Diskurs eigene und fremde Lösungsansätze zu bewerten. | | | | | |
| Inhalt: Die Komplexitätstheorie untersucht und klassifiziert Berechnungsprobleme bezüglich ihrer algorithmischen Schwierigkeit. Ziel ist es, den inhärenten Ressourcenverbrauch bezüglich verschiedener Ressourcen wie Rechenzeit oder Speicherplatz zu bestimmen, und Probleme mit ähnlichem Ressourcenverbrauch in Komplexitätsklassen zusammenzufassen. Die bekanntesten Komplexitätsklassen sind sicherlich P und NP, die die in polynomieller Zeit lösbaren bzw. verifizierbaren Probleme umfassen. Die Frage, ob P und NP verschieden sind, wird als eine der bedeutendsten offenen Fragen der theoretischen Informatik, ja sogar der Mathematik, angesehen. P und NP sind jedoch nur zwei Beispiele von Komplexitätsklassen. Andere Klassen ergeben sich unter anderem bei der Untersuchung der des benötigten Speicherplatzes, der effizienten Parallelisierbarkeit von Problemen, der Lösbarkeit durch zufallsgesteuerte Algorithmen, und der approximativen Lösbarkeit von Problemen. Die Vorlesung hat das Ziel, einen breiten Überblick über die grundlegenden Konzepte und Resultate der Komplexitätstheorie zu geben: | | | | | |
| <ul style="list-style-type: none"> • Klassische Resultate für Platz- und Zeitkomplexitätsklassen: z.B. die Korrespondenz zwischen Spielen und Speicherplatz-Beschränkungen, der Nachweis, dass sich mit mehr Platz oder Zeit auch mehr Probleme lösen lassen, weitere grundlegende Beziehungen zwischen Zeit- und Platzbasierten Klassen, und die Komplexitätswelt zwischen NP und PSPACE • Grundzüge der Komplexitätstheorie paralleler, zufallsbasierter und approximativer Algorithmen • Einführung in ausgewählte neuere Themen: Komplexitätstheorie des interaktiven Rechnens, des probabilistischen Beweisens und Fine-grained Complexity. | | | | | |

KAPITEL 1. WAHLPFLICHTMODULE

| |
|--|
| Lehrformen: Vorlesung mit begleitenden Übungen |
| Prüfungsformen: Abschlussprüfung; mündlich, 20-30min |
| Voraussetzungen für die Vergabe von Kreditpunkten: bestandene mündliche Prüfung |
| Verwendung des Moduls (in anderen Studiengängen): Studiengang Mathematik |
| Stellenwert der Note für die Endnote: 9 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Thomas Zeume |
| Sonstige Informationen: Erst wieder ab SS 22! |
| Literatur: Einstiegsliteratur für diese Veranstaltung sind die Bücher: <ul style="list-style-type: none">• Wegener. Komplexitätstheorie: Grenzen der Effizienz von Algorithmen. Springer. 2003.• Arora, Barak. Computational Complexity: A Modern Approach. Cambridge University Press. Eine Vorabversion ist verfügbar unter: http://theory.cs.princeton.edu/complexity/book.pdf• Papadimitriou. Computational Complexity. Addison-Wesley. Reading. 1995.• Kozen. Theory of Computation. Springer. 2006. |

Anwendungsbereich

| Design Optimization | | | | | |
|--|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2a | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Design Optimization (129007) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: JAVA-Grundkurs | | | | | |
| <p>Lernziele (learning outcomes):</p> <p>An important goal of this course is to present the theoretical foundations of mathematical optimization to students in a manner which allows them to use and employ design optimization for engineering applications in a sensible manner. This is achieved with a combination of theoretical lectures and practical exercises carried out using various computers software systems. In the second part of the course, students carry out team projects to solve engineering design tasks using their fundamental knowledge acquired in the first half of the course. Further, students must clearly present their projects results in a classroom setting to an audience with various technical background (the course is attended by students from applied computer science as well as by students from computational engineering). When the students have successfully complete this course, students</p> <ul style="list-style-type: none"> • will be familiar with the types of numerical algorithms available today to solve, in particular, advanced engineering tasks; • will be able to program software components to carry out design optimization tasks or employ engineering software systems to include design optimization aspects; • will have a good understanding of the basics of design optimization to be able to select proper optimization techniques in a given engineering situation and be able to implement efficient numerical solutions. | | | | | |
| Inhalt: | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- Structural optimization as a tool for the optimal design of engineering tasks with respect to given quality objective functions, side constraints as well as inequality constraints.
- Development of optimization models for use in engineering applications
- Types of optimization categories (continuous, linear/non-linear, deterministic, simulation-based, multi-level, etc.)
- Strategies of optimization methods (classic indirect methods, direct numerical methods, global evolution strategies, partial swarm methods, distributed parallel methods, etc.)
- Software systems to implement design optimization tasks
- Application of design optimization paradigms to solve engineering tasks as a team project

Lehrformen:

lectures, exercises using computers, team projects

Prüfungsformen:

Semesterbegleitend; Completion of a team project (2-4 students); presentation of project results; oral examination

Voraussetzungen für die Vergabe von Kreditpunkten:

successful completion of a team project; presentation of project results in the classroom; oral examination

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Markus König

Prof. Dr. Markus König, Dr. Karlheinz Lehner

| Design sozio-technischer Informationssysteme | | | | | |
|---|-------------------------|---------------------------|--------------------------------|-------------------------------|-----------------------------|
| Modul-Nr: 2b | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Design sozio-technischer Informationssysteme (260084) b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: 4-8 |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Mensch-Maschine-Interaktion | | | | | |
| Lernziele (learning outcomes): Die Teilnehmer/innen sind in der Lage, bei der Entwicklung von informationstechnischen Lösungen frühzeitig den späteren Anwendungskontext und die Nutzungsbedingungen zu berücksichtigen. Sie können helfen, die für die Systemeinführung und -Einsatz notwendigen organisatorischen Maßnahmen zu konzipieren. | | | | | |
| Inhalt: Die Nutzung computerbasierter Kommunikations- und Informationssysteme wird in der Regel in menschliche Handlungsabläufe eingebettet. Die Vorlesung vermittelt Kenntnisse derjenigen Faktoren und Methoden, die bei der Entwicklung und Einführung informationstechnischer Systeme dazu beitragen, dass die Nutzung erfolgreich ist. Dabei wird davon ausgegangen, dass technische, organisatorische und soziale Strukturen integriert und angepasst werden müssen. Die Erfolgsfaktoren werden aus interdisziplinärer Sicht behandelt und anhand von Beispielen aus konkreten Anwendungsfällen erläutert. Dabei werden die folgenden Gebiete berücksichtigt: | | | | | |
| <ul style="list-style-type: none"> • Arbeitswissenschaft und Ergonomie • Psychologie • Organisation und Management • Rechtliche Aspekte • Betriebswirtschaftliche Aspekte • Kommunikationstheorie • Datenschutz | | | | | |
| Es werden verschiedene Methoden vermittelt, die die Einführung von Informationssystemen unterstützen: | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- Erhebung von Anforderungen und Ausgangsbedingungen
- Usability-Engineering
- Contextual Design
- Sozio-technische Modellierung
- Partizipation und Kommunikationsmoderation

Lehrformen:

Abwechslung von Vorlesungsinhalten, Seminarbeiträgen der Studierenden, Präsenzübungen und Erörterung der Projektbearbeitung

Prüfungsformen:

Semesterbegleitend; Seminarbeitrag, Präsentation der Bearbeitung einer Projektaufgabe in Kleingruppen, Prüfungsgespräch zur Überprüfung der Eigenleistung bei der Projektarbeit

Voraussetzungen für die Vergabe von Kreditpunkten:

Ein Seminarbeitrag (25% der Endnote) und Präsentation der Projektarbeit (75%)

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann

Markus Jelonek Msc.

Sonstige Informationen:

Literatur: Th. Herrmann (2012): Kreatives Prozessdesign. Springer-Gabler.

| | | | | | |
|--|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| IT im Engineering | | | | | |
| Modul-Nr: 2c | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung IT im Engineering (133640) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| <p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verfügen Studierende über ein breites, integriertes Wissen über die Funktionalitäten und Anwendungsspezifika prozessübergreifender server- oder cloud-basierter IT Anwendungen im Engineering bzw. im gesamten Produktlebenszyklus, um Digitalisierungsprojekte in Industrieunternehmen aus Sicht des Engineerings maßgeblich vorantreiben zu können. • kennen und verstehen Studierende wesentliche Methoden und Verfahren zur systematischen Erfassung, Darstellung und Analyse von Prozessen im Engineering und zur Konzeptionierung von unterstützenden IT-Anwendungen. Indem sie praktische Fallbeispiele und Aufgaben bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit entsprechenden Softwaresystemen auf konkrete Problemstellungen der Digitalisierung von Prozessen und der Informationsverarbeitung im Engineering übertragen. • haben Studierende ein umfassendes Verständnis vom Zusammenwirken der relevanten Softwaresysteme und von Integrationskonzepten aus Sicht des Engineerings erworben und können dieses situativ angepasst anwenden, um z.B. differenziert die Eignung von IT Verfahren in den Bereichen PDM/PLM, ERP, CRM, MES zu beurteilen. • verfügen Studierende über vertiefte, interdisziplinäre Methodenkompetenz und können Aufgabenstellungen der Einführung, des Betriebs und der Integration von prozessübergreifenden IT Anwendungen und industrieller Standard-Software reflektieren und bewerten sowie selbstgesteuert verfolgen. • können Studierende kooperativ Aufgabenstellungen der Prozessmodellierung, Konzeptentwicklung und Systemspezifikation von IT Anwendungen im Engineering in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen und über Sachverhalte umfassend kommunizieren. | | | | | |
| <p>Inhalt: Die Veranstaltung vermittelt eine Übersicht über Aufgaben und Funktionsweisen von prozessübergreifenden IT-Anwendungen im Produktlebenszyklus aus den Bereichen Produktentwicklung, Produktionsplanung, Fertigung und Montage, Service und Instandhaltung, Marketing</p> | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

und Vertrieb sowie Materialwirtschaft und Logistik, immer mit Bezug zu Aufgaben des Engineerings sowie den relevanten Informationsflüssen, Schnittstellen und Integrationsaspekten. Der Schwerpunkt liegt dabei auf dem erforderlichen Grundlagenwissen in Bezug auf Software und Informationstechnologien (z.B. Datenbanken, IoT Technologien, Middleware-Konzepte sowie Cloud- und Edge Computing) und den relevanten methodischen Aspekten der Erstellung von Konzepten für die Prozessoptimierung, der Einführung von prozessunterstützenden IT-Lösungen sowie der IT-Organisation und des IT-Managements.

Lehrformen:

Vorlesung

Prüfungsformen:

Abschlussprüfung; Klausur / 90 Minuten, Anteil der Modulnote : 100 Prozent. Schriftliche Klausur mit Fragen und Aufgaben zum gesamten Stoff der Vorlesung und der Übung

Voraussetzungen für die Vergabe von Kreditpunkten:

- Studienleistung: Im Rahmen der Übung werden 5 Projektaufgaben gestellt, die bearbeitet und abgegeben werden müssen. Für die Zulassung zur Prüfung müssen mind. 4 Projektaufgaben positiv bewertet sein.
- erfolgreiche Teilnahme an der Klausur

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Maschinenbau

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr.-Ing. Detlef Gerhard

| Verkehrstechnik | | | | | |
|--|-------------------------|---------------------------|--|--|-----------------------------|
| Modul-Nr: 2d | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: | | | Kontaktzeit: | Selbststudium: | Gruppengröße: |
| <ul style="list-style-type: none"> a) Vorlesung Verkehrssteuerung b) Vorlesung Modellierung und Simulation des Verkehrsflusses | | | <ul style="list-style-type: none"> a) 2 SWS b) 2 SWS | <ul style="list-style-type: none"> a) 60h b) 60h | beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Empfohlene Vorkenntnisse: Kenntnisse in Verkehrstechnik | | | | | |
| Lernziele (learning outcomes): | | | | | |
| <ul style="list-style-type: none"> a) Die Hörer verfügen über das aktuelle technische Wissen und besitzen erweiterte Kenntnisse über die Methoden der verkehrstechnischen Analyse und Steuerung von Knotenpunkten. Sie haben die Fähigkeit, die in der Praxis angewandten Planungstechniken für Lichtsignalanlagen zu verstehen und komplexe Anlagen einschließlich einer Koordination praxisgerecht zu entwerfen. b) Die Hörer verfügen über differenzierte Kenntnisse der Gesetzmäßigkeiten des Verkehrsflusses auf Straßen. Sie sind in der Lage, wissenschaftliche Beschreibungsmöglichkeiten dieser Gesetzmäßigkeiten zu reflektieren und ihre praktische Anwendbarkeit zu erkennen. Sie haben die Fähigkeit, selbständig Erweiterungen oder Anpassungen von Verkehrsflussmodellen zu entwickeln. | | | | | |
| Inhalt: | | | | | |
| <ul style="list-style-type: none"> a) Es werden Methoden der verkehrstechnischen Analyse und Bemessung von Straßenknotenpunkten sowie Steuerungssysteme für Knotenpunkte und die zu ihrem Betrieb erforderlichen Einrichtungen behandelt. Die in der Praxis üblichen Verfahren werden in der Übung an einigen Beispielen veranschaulicht. Dabei werden EDV-Verfahren eingesetzt. Im Einzelnen werden behandelt: Wartezeitermittlung an Knotenpunkten, vorfahrtgeregelte Knotenpunkte, Festzeitsteuerung von Signalanlagen, Grüne Welle, Koordination im Netz, verkehrsabhängige Steuerung einschließlich Signalprogrammgebung, Signaltechnik, Steuerungskriterien. b) Die theoretischen Grundlagen für die Beschreibung des Verkehrsflusses auf Straßen werden mit Hilfe mathematischer Verfahren erarbeitet. Die zu Grunde liegenden Gesetzmäßigkeiten werden hergeleitet. Im Einzelnen werden behandelt: Kenngrößen des Verkehrsablaufs und deren Zusammenhänge, Fundamentaldiagramm, Kapazität, freier Verkehrsfluss, Kontinuumstheorie, Abstandsmodelle, Fahrzeugfolgetheorie, mikroskopische Verkehrsflusssimulation. | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|--|
| Lehrformen: Powerpoint-Präsentationen, Tafel, Vorführungen und Übungen am PC |
| Prüfungsformen: Abschlussprüfung; Klausur / 120 Minuten; Klausur über das gesamte Modul |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Bauingenieurwesen |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Justin Geistefeldt a) Apl. Prof. Dr.-Ing. Ning Wu b) Prof. Dr.-Ing. Justin Geistefeldt |
| Sonstige Informationen: Literatur: Ausführliches Skript zur Lehrveranstaltung; Schnabel, Lohse: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung, Band 1 - Verkehrstechnik, Beuth Verlag Einschlägige Richtlinien und Merkblätter (werden in der Vorlesung genannt) |

| Grundlagen der automatischen Spracherkennung | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2e | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Grundlagen der automatischen Spracherkennung (141044) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: | | | | | |
| <ul style="list-style-type: none"> • Grundkenntnisse der digitalen Signalverarbeitung und der Wahrscheinlichkeitsrechnung • Grundlegende Programmierkenntnisse | | | | | |
| Lernziele (learning outcomes): Die Teilnehmer verstehen die theoretischen und praktischen Grundlagen automatischer Spracherkennungssysteme. Sie sind in der Lage, die Kernalgorithmen eines einfachen Spracherkenners selbstständig zu implementieren und verstehen die Prinzipien von aktuellen Erkennungssystemen für kleines und großes Vokabular. Dabei wird auch ein Verständnis für die Entwicklung von automatischen Mustererkennungsverfahren für ein breites Anwendungsfeld entwickelt. | | | | | |
| Inhalt: Die Vorlesung vermittelt Grundlagen und Algorithmen der maschinellen Spracherkennung in der Form, in der sie in aktuellen Systemen zur Erkennung fließender Sprache eingesetzt werden. Die folgenden Themen werden behandelt: | | | | | |
| <ul style="list-style-type: none"> • Grundlagen: Phonetik, Sprachwahrnehmung • Statistische Methoden: Klassifikation, Schätztheorie • Klassifikation mittels Deep Neural Networks • Merkmalsextraktion: Merkmale im Zeit- und Frequenzbereich, Cepstralanalyse • Spracherkennung mit Hidden Markov Modellen: Algorithmen, Modellinitialisierung, Baum-Welch-Reestimation, Numerische Aspekte, Systeme zur Einzel- und Verbundwörtererkennung, HMM/DNN-Systeme Gleichzeitig werden in einem Programmierpraktikum die eingeführten Methoden angewandt. | | | | | |
| Die Übung ist projektorientiert; alle Übungsaufgaben zusammengenommen ergeben einen Verbundwörterkennner für fließend gesprochene Ziffernkettensätze. Dieser wird in Arbeitsgruppen von 2-3 Studenten erarbeitet. | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Lehrformen: Vorlesung und Praxisübungen |
| Prüfungsformen: Abschlussprüfung; Klausur schriftlich, 120 Minuten |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Elektrotechnik |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Dorothea Kolossa |

| Simulationstechnik | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2f | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Simulationstechnik (127012) b) Übung | | | Kontaktzeit: 3 SWS | Selbststudium: 105 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Objektorientierte Modellierung | | | | | |
| Lernziele (learning outcomes): Dieses Modul vermittelt Kompetenzen zum Einsatz von rechnergestützten Simulationskonzepten zur Lösung praxisbezogener Aufgabenstellungen im Bau- und Umweltingenieurwesen. Im Rahmen der Übungsveranstaltungen erfolgt eine Einführung in aktuelle Simulations- und Optimierungssoftware. Die Projektarbeit wird als Gruppenarbeit durchgeführt. | | | | | |
| Inhalt: Es werden Vorgehensweisen zur Simulation komplexer Systeme vermittelt. Folgende Themen werden behandelt: <ul style="list-style-type: none"> • Systemanalyse und Modellbildung • System Dynamics • Ereignisdiskrete Simulation • Multiagentensimulation • Aufbereitung von Eingangsdaten • Stochastische Simulation • Simulationsgestützte Optimierung • Einführung in die Software AnyLogic <p>Innerhalb der Projektarbeit werden aktuelle Fragestellungen aus den Bau- und Umweltingenieurwissenschaften (Montage- und Logistikprozesse, Fußgängersimulation, Schadstoffausbreitung, etc.) aufgearbeitet und mit Hilfe einer objekt-orientierten Simulationssoftware analysiert. Den Studierenden werden Softwarelizenzen durch den Lehrstuhl zur Verfügung gestellt. Die Programmierung erfolgt unter Verwendung der Programmiersprache Java.</p> | | | | | |
| Lehrformen: Vorlesung und Übung | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Prüfungsformen: Semesterbegleitend; Projektarbeit |
| Voraussetzungen für die Vergabe von Kreditpunkten: erfolgreiche Projektarbeit |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Bauingenieurwesen, M.Sc. Umwelttechnik und Ressourcenmanagement |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Markus König |
| Sonstige Informationen: Literatur: <ol style="list-style-type: none">1. Banks, J.; Carson II, J. S.; Nelson, B. L.; Nicol, D. M. (2005): Discrete-Event System Simulation, Pearson Prentice Hall2. Bossel, H. (1994): Modellbildung und Simulation : Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme, ein Lehr- und Arbeitsbuch, Vieweg Verlag3. Biethahn, J.; Lackner, A.; Range, M; Brodersen, O. (2004): Optimierung und Simulation, Oldenbourg Verlag, München4. Simulationssoftware AnyLogic der Firma XJ Technologies, http://www.anylogic.com |

| 3D-Simulation in der Automatisierungstechnik | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|--|
| Modul-Nr: 2g | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung 3D-Simulation in der Automatisierungstechnik (139050) | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: 20 Studierende (Übungen) |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Besuch der Vorlesung "Grundlagen der Automatisierungstechnik" | | | | | |
| <p>Lernziele (learning outcomes): Nach einer allgemeinen Einführung in die Anforderungen und Möglichkeiten der 3D-Simulation in der Automatisierungstechnik erlernen die Studierenden die verschiedenen Schritte, die sich von der Idee über die Modellierung der Arbeitszelle und Programmierung der Automatisierungskomponenten bis zur Virtuellen Inbetriebnahme erstrecken. Ein besonderes Augenmerk wird in dieser Vorlesung auf die Industrielle Robotik gelegt, die in zahlreichen Beispielen und Anwendungen thematisiert wird. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • beherrschen die Studierenden die Grundlagen der 3D-Simulationstechnik. • haben die Studierenden Fähigkeit erworben, automatisierungstechnische Aufgabenstellungen vorab über eine 3D-Simulation abzubilden und abzusichern. • kennen die Studierenden die wichtigsten Methoden und Softwaresysteme zur Lösung simulationstechnischer Probleme. | | | | | |
| <p>Inhalt: Die Vorlesung deckt die folgenden Themenbereiche ab</p> <ul style="list-style-type: none"> • Simulation in der Automatisierungstechnik: Anforderungen und Möglichkeiten • Grafische 3D-Simulation • CAD-basierte Arbeitszellenmodellierung und 3D-Datenaustausch • Roboterprogrammierung • Offline-Programmierung und Virtuelle Inbetriebnahme • Grundlagen und Leistungsmerkmale von grafischen 3D-Simulationssystemen im industriellen Einsatz | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

Im Bereich der Kompetenzen im Kontext Digitalisierung ist besonders die Übung zu nennen, diese besteht aus der praktischen Umsetzung der von Modellierungs-, Programmierungs- und Simulationsaufgaben mit einem kommerziell verfügbaren und industriell eingesetzten 3D-Robotersimulations- und Offline-Programmiersystem.

Lehrformen:

Gemeinsame Vorlesung und Übung in kleineren Gruppen

Prüfungsformen:

Abschlussprüfung; Klausur (90 Minuten)

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussprüfung

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Maschinenbau

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr.-Ing. Alfred Hypki

Sonstige Informationen:

Bei geringer Teilnehmerzahl kann die Prüfung auch mündlich (30 min.) angeboten werden.

| Complexity Economics and Agent-based Modeling | | | | | |
|---|--------------------------|---------------------------|--------------------------------|--------------------------------|---|
| Modul-Nr: 2h | Credits: 10 CP | Workload: 300 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Complexity Economics and Agent-based Modeling (075240) b) Übung (075246) | | | Kontaktzeit: 4 SWS | Selbststudium: 240 h | Gruppengröße: 30 Studierende aus den Wirtschaftswissenschaften und der Angewandten Informatik |
| Sprache: English | | | | | |
| Teilnahmevoraussetzungen: Gute Englischkenntnisse Grundlegende Programmierkenntnisse Empfohlene Vorkenntnisse: Successful completion of a “Studienleistung”. Macroeconomics I and Macroeconomics II strongly recommended. Willingness to learn computer programming. | | | | | |
| <p>Lernziele (learning outcomes): Das Modul verfolgt das Ziel, Studierende der Angewandten Informatik sowie der Wirtschaftswissenschaft in die Methode der agentenbasierten Modellierung und den dazugehörigen Simulationstechniken einzuführen. Hierbei lernen Studierende wie volkswirtschaftliche und sozialwissenschaftliche Fragestellungen entworfen, implementiert und analysiert werden. Im Rahmen der Seminararbeit sollen die Studierenden ein eigenes Modell implementieren und analysieren.</p> <p>Im Rahmen des Moduls erwerben Studierende folgende Kenntnisse:</p> <ol style="list-style-type: none"> 1) Grundlegende Kenntnisse der Komplexitätsökonomik und agentenbasierter Modellierung 2) Überführung von wissenschaftlichen Fragestellungen aus dem Bereich Volkswirtschaftslehre in agentenbasierte Computersimulationen 3) Praktische Arbeit mit agentenbasierten Modellen, Interpretation von Simulationsergebnissen und mögliche Grenzen des Ansatzes 4) Grundlegende Kenntnisse, um eigene Modelle zu implementieren und Simulationen selbstständig durchzuführen 5) Wissenschaftliches Schreiben 6) Die Programmiersprache NetLogo | | | | | |
| <p>Inhalt: Komplexitätsökonomik ist eine neue Denkschule im Bereich der Volkswirtschaftslehre, der zunehmend an Relevanz gewinnt. Hierbei wird eine Volkswirtschaft als ein komplexes, adaptives System betrachtet, das sich aus verschiedenen Agenten, z.B. Firmen, Haushalten</p> | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

und Banken, zusammensetzt. Im Vergleich zu bisherigen volkswirtschaftlichen Ansätzen sind die Agenten des Systems nicht (vollständig) rational und das System entwickelt sich aus den Interaktionen zwischen den (heterogenen) Agenten. Komplexitätsökonomik eignet sich sehr zur Untersuchung von sozialen Transformations- und Innovationsprozessen, wie der Digitalisierung.

In der Vorlesung wird Komplexitätsökonomik und die Methode der agentenbasierten Computersimulation vorgestellt und gezeigt, wie sie zur Analyse komplexer ökonomischer Systeme angewendet werden können. Im Rahmen der Übung werden diese Kenntnisse vertieft. Zudem wird geübt, wie agentenbasierte Modelle mithilfe NetLogo implementiert und simuliert werden.

Lehrformen:

Vorlesung und Übung

Prüfungsformen:

Semesterbegleitend; Seminararbeit, Klausur

Voraussetzungen für die Vergabe von Kreditpunkten:**Bestehen der folgenden Leistungen:**

- 1) Studienleistung (unbenotet)
- 2) Seminararbeit (70%) und Klausur (30%) (benotet)

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Economic Policy Consulting, M.Sc. Economics, M.Sc. Management and Economics, M.Sc. Management, M.Sc. Sales Management

Stellenwert der Note für die Endnote: 10 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Michael Roos

Sonstige Informationen:

Verbindliche Anmeldung in der ersten Sitzung und durch Registrierung bei Moodle. Studierende der angewandten Informatik müssen sich beim Zentrum für ökonomische Bildung (ZfÖB) zum Modul anmelden, um eine Prüfungsleistung erbringen zu können.

Literatur: Will be announced at the beginning of the module.

| High-Performance Computing on Clusters | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2i | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung High-Performance Computing on Clusters b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): In this module, the students acquire professional skills to program and employ parallel computing clusters. Theoretical properties of distributed-memory systems and programming patterns are conveyed as well as the practical implementation. | | | | | |
| Inhalt: The lecture deals with the parallelization on cluster computers. Distributed-memory programming concepts (MPI) are introduced and best-practice implementation is presented based on applications from scientific computing including the finite element method and machine learning. Special attention is paid to scalable solvers for systems of equations on distributed memory systems, focusing on iterative schemes such as simple splitting methods (Richardson, Jacobi, Gauß-Seidel, SOR), Krylov-methods (Gradient descent, CG, BiCGStab) and, in particular, the multigrid method. The mathematical foundations for iterative solvers are reviewed, suitable object-oriented interface structures are developed and an implementation of these solvers for modern parallel computer architectures is developed. Numerical experiments and self-developed software implementations are used to discuss and illustrate the theoretical results. | | | | | |
| Lehrformen: Beamer, computer lab, numerical experiments | | | | | |
| Prüfungsformen: Abschlussprüfung; Klausur / 120 Minuten | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Bauingenieurwesen, M.Sc. Computational Engineering | | | | | |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |

Modulbeauftragte/r und hauptamtlich Lehrende:

Jun. Prof. Dr. Andreas Vogel

Sonstige Informationen:

Literatur:

- W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, Springer, 1994
- Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003
- MPI Application Programming Interface (2015), www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf
- W. Gropp, E. Lusk, A. Skjellum, Using MPI, MIT Press, 2014
- S. Snir, S. Otto. S. Huss-Lederman, D. Walker, J. Dongarra, MPI – The Complete Reference, MIT Press, 1998
- T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer, 2013
- C. Douglas, G. Haase, U. Langer, A Tutorial on Elliptic PDE Solvers and Their Parallelization, SIAM, 2003
- additional literature will be announced in the lecture

| High-Performance Computing on Multi- and Manycore Processors | | | | | |
|--|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2j | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung High-Performance Computing on Multi- and Manycore Processors b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): In this module, the students acquire professional skills to program multi- and manycore processors employing multi-threaded execution and handling shared-memory access patterns. Theoretical properties are conveyed as well as practical implementation. Via presentations of selected topics, students attain the ability to survey and acquire knowledge on advanced scientific topics independently and are qualified to illustrate such topics in the form of a presentation and numerical examples. | | | | | |
| Inhalt: The lecture addresses parallelization for multi- and manycore processors. Thread-based programming concepts (pthreads, C++11 threads, OpenMP, OpenCL) are introduced and best-practice implementation aspects are highlighted based on applications from scientific computing. In the first part, the lecture provides an overview on relevant data structures, solver techniques and programming patterns from scientific computing. An introduction to multi-threading programming on multicore systems is then provided with special attention to shared-memory aspects. Parallelization patterns are discussed and highlighted. Numerical experiments and self-developed software implementations are used to discuss and illustrate the presented content. In the second part, students are assigned advanced topics for shared-memory computation from the engineering science including finite element methods and artificial intelligence. Based on a scientific paper, students present their topic to the lecture audience in form of a beamer presentation and numerical illustrations. | | | | | |
| Lehrformen: Beamer, computer lab, numerical experiments | | | | | |
| Prüfungsformen: Semesterbegleitend; Hausarbeit | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: mit mindestens ausreichend bewertete Hausarbeit | | | | | |

| |
|---|
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Bauingenieurwesen, M.Sc. Computational Engineering |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Jun. Prof. Dr. Andreas Vogel |
| Sonstige Informationen: Literatur: <ul style="list-style-type: none">• G. Hager, G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010• T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer, 2013• OpenMP Application Programming Interface (2015), www.openmp.org/wp-content/uploads/openmp-4.5.pdf• OpenCL Application Programming Interface (2014), www.khronos.org/registry/OpenCL/specs/opencl-1.2.pdf• additional literature will be announced in the lecture |

| Knowledge Graphs | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2k | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Knowledge Graphs (150336) b) Übung (150337) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: English | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Basic knowledge about the following topics is highly recommended but not mandatory: Graph theory, set theory, databases, logic. | | | | | |
| Lernziele (learning outcomes): In this lecture, students will learn about the foundations of modelling, querying, publishing, and reasoning over KGs. The topics will be complemented with exercises and Jupyter Notebooks (https://jupyter.org/) to show how KG technologies work in practice. | | | | | |
| Inhalt: Knowledge Graphs (KG) allow for representing inter-connected facts or statements annotated with semantics. In KGs, concepts and entities are typically modeled as nodes while their connections are modeled as directed and labeled edges, creating a graph. In recent years, KGs have become core components of modern data ecosystems. KGs, as building blocks of many Artificial Intelligence approaches, allow for harnessing and uncovering patterns from the data. Currently, KGs are used in the data-driven business processes of multinational companies like Google, Microsoft, IBM, eBay, and Facebook. Furthermore, thousands of KGs are openly available on the web following the Linked Data (https://lod-cloud.net/) principles. The specific topics covered in the lecture are as follows: <ol style="list-style-type: none"> 1. Introduction to Knowledge Graphs 2. The Resource Description Framework (RDF) 3. RDF Schema (RDFS) 4. The SPARQL Query Language 5. Semantics of SPARQL 6. Linked Data: Knowledge Graphs and Ontologies on the Web 7. The Web Ontology Language (OWL) 8. Entailment Regimes 9. Reasoning over Knowledge Graphs | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| 10. Property Graphs |
| 11. Knowledge Graph Applications |
| Lehrformen: Vorlesung mit begleitenden Übungen |
| Prüfungsformen: Written exam |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): Studiengang Mathematik |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Jun.-Prof. Maribel Acosta Deibe |
| Sonstige Informationen: Literatur: <ul style="list-style-type: none">• Aidan Hogan et al. Knowledge Graphs. 2020. https://arxiv.org/pdf/2003.02320.pdf• Andreas Harth. Introduction to Linked Data. (Specific chapters will be provided in the lecture).• Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman and Hall/CRC, 2009. |

KAPITEL 2. ANWENDUNGSBEREICH

| Autonomous Robotics: Action, Perception and Cognition | | | | | |
|--|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 21 | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Autonomous Robotics: Action, Perception and Cognition (310501) b) Übung (310511) | | | Kontaktzeit: 3 SWS | Selbststudium: 135 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): After the successful completion of this course the students: <ul style="list-style-type: none"> • are familiar with the concepts in dynamical systems theory and can use them practically, • know the mathematical models of movement generation • have practice in reading and writing academic research papers. | | | | | |
| Inhalt: Autonomous robotics is an interdisciplinary research field in which embodied systems equipped with their own sensors and with actuators generate behavior that is not completely pre-programmed. Autonomous robotics thus entails perception, movement generation, as well as core elements of cognition such as making decisions, planning, and integrating multiple constraints. This course touches on various approaches to this interdisciplinary problem. In the first half of the course, the main emphasis will be on dynamical systems methods for generating movement in vehicles. The main focus of the course is, however, on solutions to autonomous movement generation that are inspired by analogies with how nervous systems generate movement. In fact, the second half of the course will review core problems in human movement science, including the degree of freedom problem, coordination, motor control, and the reflex control of muscles. | | | | | |
| Lehrformen: Vorlesung mit Übung | | | | | |
| Prüfungsformen: Abschlussprüfung; oral exam; Exercises for bonus-points | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: passed Oral exam | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Elektrotechnik und Informationstechnik | | | | | |

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Gregor Schöner

Sonstige Informationen:

Literatur:

1. Valentino Braitenberg: Vehicles. Experiments in Synthetic Psychology, MIT Press, Cambridge, Mass 1984
2. Gregor Schöner, Michael Dose, Christoph Engels: Dynamics of behavior: Theory and applications for autonomous robotic architectures. Robotics and Autonomous Systems, 16:213-245 (1995)
3. Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, and Gregor Schöner: Dynamic Neural Fields as Building Blocks of a Cortex-Inspired Architecture for Robotic Scene Representation. IEEE Transactions Autonomous Mental Development 3:74-91 (2011)

| Computational Neuroscience: Neural Dynamics | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2m | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Computational Neuroscience: Neural Dynamics (310001) b) Übung (310011) | | | Kontaktzeit: 3 SWS | Selbststudium: 135 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): <ul style="list-style-type: none"> • Gain experience in interdisciplinarity bridging computer science and cognitive science. • Learn the concepts and methods of nonlinear dynamical systems in a concrete applied context. • Improve familiarity with methods of quantitative natural science, including measurement, graphing observables as a function of experimental control parameters and using models to interpret data. • Read scientific literature. | | | | | |
| Inhalt: <p>This course provides an introduction into the theoretical cognitive and functional neurosciences from a particular theoretical vantage point, the dynamical systems approach. This approach emphasizes the evolution in time of behavioral and neural patterns as the basis of their analysis and synthesis. Dynamic stability, a concept shared with the classical biological cybernetics framework, is one cornerstone of the approach. Instabilities (or bifurcations) extend this framework and provide a basis for understanding flexibility, task specific adjustment, adaptation, and learning. The course includes tutorial modules that provide mathematical foundations. Theoretical concepts are expounded in reference to a number of experimental model systems which include the coordination of movement, postural stability, the perception of motion, and elementary forms of embodied cognition. In the spirit of Braitenberg's "synthetic psychology", autonomous robots are used to illustrate some of the ideas. Exercises are integrated into the lectures. They consist of elementary mathematical exercises, the design of (thought) experiments and their analysis, and the design of simple artificial systems, all on the basis of the theoretical framework exposed in the main lectures. One exercise takes the form of an essay for which participants read a scientific paper and answer questions in a longer illustrated text.</p> | | | | | |
| Lehrformen: Vorlesung mit Übung | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Prüfungsformen: Abschlussprüfung; Klausur oder mündliche Prüfung |
| Voraussetzungen für die Vergabe von Kreditpunkten: Passed written or oral examination; Bonus points for the final examination can be achieved by submitting homework and an essay (10 pages). |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Cognitive Science |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Gregor Schöner |
| Sonstige Informationen: Literatur: <ol style="list-style-type: none">1. Martin Braun: Differential equations and their applications, Springer Verlag, New York, 19932. Gregor Schöner and Scott Kelso: Dynamic Pattern Generation in Behavioral and Neural Systems. Science 239: 1513-1520 (1988)3. Gregor Schöner: Dynamical Systems Approaches to Cognition. In: The Cambridge Handbook of Computational Psychology,4. Ron Sun, (ed.), Cambridge University Press (2008), pages 101-126 |

| Computational Neuroscience: Vision and Memory | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|-----------------------------|
| Modul-Nr: 2n | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Computational Neuroscience: Vision and Memory (310504) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: 20 |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: The mathematical level of the course is mixed but generally high. The tutorial is almost entirely mathematical. Mathematics required include calculus (functions, derivatives, integrals, differential equations, ...), linear algebra (vectors, matrices, inner product, orthogonal vectors, basis systems, ...), and a bit of probability theory (probabilities, probability densities, Bayes' theorem, ...). | | | | | |
| Lernziele (learning outcomes): After the successful completion of this course the students: <ul style="list-style-type: none"> • know basic neurobiological facts about the visual system and the hippocampus, • know a number of related models and methods in computational neuroscience, • understand the mathematics of these methods, • can communicate about all this in English. | | | | | |
| Inhalt: This lecture covers basic neurobiology and models of selforganization in neural systems, in particular addressing Learning and self-organization <ul style="list-style-type: none"> • Hebbian Learning • Neural learning dynamics and constrained optimization • Dynamic field theory Vision <ul style="list-style-type: none"> • Receptive fields • Neural maps • Hippocampus • Navigation | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- Episodic memory
- Hopfield Network

Lehrformen:

This course is given with the flipped/inverted classroom concept. First, the students work through online material by themselves. In the lecture time slot we then discuss the material, find connections to other topics, ask questions and try to answer them. In the tutorial time slot the newly acquired knowledge is applied to analytical exercises and thereby deepened. I encourage all students to work in teams during self-study time as well as in the tutorial.

Prüfungsformen:

The course is concluded with a digital written exam.

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussprüfung

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Cognitive Science

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Laurenz Wiskott

| Master-Praktikum: Autonomous Robotics | | | | | |
|---|-------------------------|--------------------------|--------------------------------|----------------------------------|----------------------------------|
| Modul-Nr: 2o | Credits: 3 CP | Workload: 90 h | Semester: 1.-3. Sem. | Turnus: Jedes Semester | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Master-Praktikum (310036 310536) | | | Kontaktzeit: 2 SWS | Selbststudium: 60 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Vorlesung Autonomous Robotics: Action, Perception, and Cognition | | | | | |
| Lernziele (learning outcomes): <ul style="list-style-type: none"> • Einüben des praktischen Arbeitens im Umfeld der autonomen Robotik, insbesondere der Entwurf von mathematischen Verfahren und ihre algorithmische Implementation zur Erzeugung von zeitlich strukturierten Abläufen • Einüben der Dokumentation von technischen Programmieraufgaben. • Verständnis der Grundlagen der autonomen Robotik anhand des einfachsten Robotersystems, eines autonomen Kleinstvehikels. • Einüben des Lesens und Schreibens von wissenschaftlich-technischen Artikeln | | | | | |
| Inhalt: Ziel des Praktikums ist es, die Grundlagen der autonomen Robotik anhand von Experimenten zu vermitteln, die eine enge Kopplung an aktuelle wissenschaftliche Probleme dieses Feldes aufweisen. Am Beispiel der Kleinstroboter vom Typ E-Puck wird hierbei die Erzeugung und Organisation von künstlichem Verhalten behandelt. Unter Verwendung der robotereigenen Sensorik werden experimentelle Aufgaben durch Implementierungen in der Programmierumgebung MATLAB gelöst. Der theoretische Hintergrund der Lösungen ist der dynamische Systeme Ansatz der kognitiven Robotik. Eine optionale Erweiterung des Praktikums durch das der dritte CP erzielt wird, ist das Lesen einer wissenschaftlichen Veröffentlichung im Umfeld der im Praktikum behandelten Methoden und das Schreiben eines Essays, in dem Fragen zu dem Artikel durch einen längeren, strukturierten und illustrierten Text beantwortet werden. | | | | | |
| Lehrformen: Praktikum/Projektarbeit | | | | | |
| Prüfungsformen: Blockpraktikum; praktische Prüfung und schriftliche Dokumentation | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Voraussetzungen für die Vergabe von Kreditpunkten: erfolgreiche Bearbeitung der Programmieraufgaben; Abgabe von schriftlichen Dokumentationen |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Elektrotechnik und Informationstechnik |
| Stellenwert der Note für die Endnote: 3 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Gregor Schöner Mitarbeiter des Instituts |
| Sonstige Informationen: Da die Veranstaltung als Blockpraktikum in der vorlesungsfreien Zeit zum Ende eines Semesters stattfindet, ist die Prüfung bis 14 Tage vor Vorlesungsende anzumelden. Literatur: <ol style="list-style-type: none">1. Valentino Braitenberg: Vehicles. Experiments in Synthetic Psychology, MIT Press, Cambridge, Mass 19842. Estela Bicho and Gregor Schöner: The dynamical approach to autonomous robotics demonstrated on a low-level vehicle platform. Robotics and Autonomous Systems 21:23-35 (1997) |

KAPITEL 2. ANWENDUNGSBEREICH

| | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Deep Learning | | | | | |
| Modul-Nr: 2p | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Deep Learning (150332) b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): Die Vorlesung hat das Ziel, einen Einblick in dieses Gebiet zu vermitteln. Zu Beginn werden die grundlegenden Begriffe und Konzepte des maschinellen Lernens eingeführt. Im weiteren Verlauf wird auf verschiedene neuronale Netze, Gradienten-basierte Optimierungsverfahren und generative Modelle eingegangen. | | | | | |
| Inhalt: Deep Learning ist ein Untergebiet des maschinellen Lernens, welches in den letzten Jahren zu Durchbrüchen in zahlreichen Anwendungsgebieten (wie z.B. in der Objekt- und Spracherkennung und der maschinellen Übersetzung) geführt hat. Deep Learning Methoden finden unter anderem Anwendung im Bereich IT Security | | | | | |
| Lehrformen: Folien, Tafelanschrieb | | | | | |
| Prüfungsformen: Abschlussprüfung; schriftlich, 120 Minuten | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme | | | | | |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |
| Modulbeauftragte/r und hauptamtlich Lehrende: Jun. Prof. Dr. Asja Fischer | | | | | |

| | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse) | | | | | |
| Modul-Nr: 2q | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse) (150314) b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Inhalte der Vorlesungen Einführung in die Kryptographie 1 und 2. | | | | | |
| Lernziele (learning outcomes): Die Studierenden beherrschen die grundlegenden Algorithmen in der Kryptanalyse. | | | | | |
| Inhalt: Die Vorlesung gibt einen Einblick in grundlegende Methoden der Kryptanalyse. Der Stoffplan umfasst die folgenden Themen: <ul style="list-style-type: none"> • Brute Force und Geburtstagsangriffe • Time-Memory Tradeoffs • Seitenkanalangriffe • Gittertheorie und der LLL-Algorithmus • Gitterbasierte Angriffe auf RSA • Hidden Number Problem und Angriffe auf DSA • Faktorisieren mit Faktorbasen • Diskreter Logarithmus, Index-Calculus | | | | | |
| Lehrformen: Vorlesung und Übung | | | | | |
| Prüfungsformen: Abschlussprüfung; Klausur (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|--|
| Verwendung des Moduls (in anderen Studiengängen): B.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme, B.Sc. Mathematik, M.Sc. Mathematik |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Alexander May |
| Sonstige Informationen: Literatur: Skript zur Vorlesung. |

KAPITEL 2. ANWENDUNGSBEREICH

| Symmetrische Kryptanalyse | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2r | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Symmetrische Kryptanalyse (150351) b) Übungen | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Einführung in die Kryptographie 1 | | | | | |
| Lernziele (learning outcomes): Die Studierenden haben ein vertieftes Verständnis für die Sicherheit symmetrischer Chiffren. | | | | | |
| Inhalt: Wir behandeln die wichtigsten Themen in der symmetrischen Kryptanalyse. Nach einer ausführlichen Vorstellung von linearer und differentieller Kryptanalyse werden weitere Angriffe auf symmetrische Primitive, insbesondere Block-Chiffren behandelt. Hierzu zählen insbesondere Integral (auch Square) Attacks, Impossible Differentials, Boomerang-Angriffe und Slide-Attacks. Neben den Angriffen selbst werden auch immer die daraus resultierenden Design-Kriterien beschrieben, um neue Algorithmen sicher gegen die Angriffe zu machen. | | | | | |
| Lehrformen: Vorlesungen und Übungen | | | | | |
| Prüfungsformen: Abschlussprüfung; mündlich, 30 Minuten | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene mündliche Prüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme, B.Sc. Mathematik, M.Sc. Mathematik | | | | | |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Gregor Leander | | | | | |
| Sonstige Informationen: Literatur: Knudsen, Lars, Robshaw, Matthew "The Block Cipher Companion", Springer, 2012 | | | | | |

| Kryptographische Protokolle | | | | | |
|---|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2s | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Kryptographische Protokolle (150343) b) Übung (150344) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Inhalte des Moduls Kryptographie | | | | | |
| Lernziele (learning outcomes): Die Studierenden verstehen die erweiterten mathematischen Methoden und Verfahren, auf denen moderne kryptographische Protokolle beruhen. Die Teilnehmer sind zur Analyse und dem Design aktueller und zukünftiger kryptographischer Methoden befähigt. | | | | | |
| Inhalt: Die Vorlesung beschäftigt sich mit erweiterten kryptographischen Protokollen und deren Anwendungen. Themenübersicht: <ul style="list-style-type: none"> • Identity-based Encryption • Digital Signatures • Secret sharing • Threshold Cryptography • Secure Multiparty Computation | | | | | |
| Lehrformen: Vorlesung (+Übung) | | | | | |
| Prüfungsformen: Abschlussprüfung; Klausur (120 Minuten) | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme, B.Sc. Mathematik, M.Sc. Mathematik | | | | | |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Eike Kiltz

| Quantenalgorithmen | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2t | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Quantenalgorithmen (150318) b) Übung | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Kenntnisse in Mathematik und Informatik | | | | | |
| <p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • beherrschen die Studierenden die Grundlagen der Quantenalgorithmen • haben die Studierenden erlernt, inwiefern sich Quantencomputer von klassischen Rechnern unterscheiden und welche Klassen von Problemen effizient von Quantenrechnern gelöst werden • werden die Studierenden in die Lage versetzt, einfache Probleme mit Quantenalgorithmen zu lösen und die Korrektheit ihrer Algorithmen nachzuweisen • können die Studierenden Quantenschaltkreise mit Python auf einem Quantenrechner oder einem Quanten-Simulator implementieren | | | | | |
| <p>Inhalt: Die Vorlesung gibt einen Einblick in die Konstruktion von Algorithmen für Quantenrechner und behandelt die folgenden Themen:</p> <ul style="list-style-type: none"> • Quantenbits und Quantengatter • Separabilität und Verschränkung • Teleportation • Quantenschlüsselaustausch • Quantenkomplexität • Simons Problem • Shors Faktorisierungsalgorithmus • Grovers Suchalgorithmus | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Lehrformen: Hörsaalvorlesung mit Medienunterstützung, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben |
| Prüfungsformen: Abschlussprüfung; Schriftliche Modulabschlussprüfung über 120 Minuten |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): B.Sc. Informatik, M. Sc. IT-Sicherheit/Informationstechnik, M.Sc. IT-Sicherheit/Netze und Systeme, B.Sc. Mathematik, M.Sc. Mathematik |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Alexander May |
| Sonstige Informationen: Zur Vorlesung existiert ein Skript. Literatur: 1. N. D. Mermin: „Quantum Computer Science – An Introduction“, Cambridge University Press, 2007 2. M. A. Nielsen, I. L. Chuang: “Quantum Computation and Quantum Information“, Cambridge University Press, 2007 3. M. Homeister: „Quantum Computing Verstehen“, Springer Verlag, 2013 |

KAPITEL 2. ANWENDUNGSBEREICH

| Systemsicherheit | | | | | |
|---|-----------------|------------------|------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: | Credits: | Workload: | Semester: | Turnus: | Dauer: |
| 2u | 5 CP | 150 h | 1.-3. Sem. | jedes SoSe | 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Systemsicherheit (141340) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): Die Studierenden beherrschen theoretische und praktische Aspekte der Sicherheit von Softwaresystemen und sind zu einer kritischen Betrachtung der Systemsicherheit in der Lage. Insbesondere erwerben die Studierenden die Fähigkeit zum Modellieren konkreter Fragestellungen und Anforderungsanalysen aus vorhandenen Systeminformationen bzw. Systemgegebenheiten. Sie können eigenständig neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Die Studierenden sind in der Lage, neue Sicherheitsmodelle selbst zu erstellen und diese argumentativ zu verteidigen. | | | | | |
| Inhalt: Im Rahmen der Vorlesung werden wichtige theoretische und praktische Aspekte aus dem Bereich der Systemsicherheit vorgestellt und diskutiert. Der Fokus liegt dabei auf verschiedenen Aspekten der Softwaresicherheit und verschiedene Angriffs- und Verteidigungstechniken werden vorgestellt. Die Studierenden sollen am Ende der Vorlesungsreihe in die Lage sein, die Sicherheit verschiedener Softwaresysteme zu analysieren, Schwachstellen im Design und der Implementierung aufzudecken sowie selbständig Sicherheitsmechanismen zu entwickeln. Darüber hinaus werden auch andere Aspekte aus dem Bereich der Systemsicherheit wie Privatheit und Anonymität betrachtet. | | | | | |
| Lehrformen: b) Übung | | | | | |
| Prüfungsformen: Schriftliche Modulabschlussprüfung | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu | | | | | |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) | | | | | |
| Modulbeauftragte/r und hauptamtlich Lehrende: | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

Prof. Dr. Thorsten Holz

Prof. Dr. Thorsten Holz, M. Sc. Thorsten Eisenhofer, M. Sc. Moritz Schlögel

| Model Checking | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2v | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Model Checking (150324) b) Übung (150325) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: <ul style="list-style-type: none"> • Grundlagenvorlesungen Mathematik, • Einführung in die Theoretische Informatik (ggf. kann das nötige Wissen auch nachgeholt werden), • Hilfreich: Logik in der Informatik, Datenstrukturen und elementare Programmierkenntnisse | | | | | |
| Lernziele (learning outcomes): Die Studierenden lernen wie sich verteilte Systeme durch Transitionssysteme modellieren und Eigenschaften in logischen Spezifikationsprachen wie LTL und CTL spezifizieren lassen. Sie sollen elementare Algorithmen zur Überprüfung von Eigenschaften in Transitionssystemen kennenlernen. Sie sollen ein Verständnis für die Möglichkeiten und Grenzen des Model Checking entwickeln, und in die Lage versetzt werden, sich eigenständig mit fortgeschrittenen Methoden des Model Checkings auseinanderzusetzen. | | | | | |
| Inhalt: Wie kann die Korrektheit von Software und Hardware formal überprüft werden? Im Model Checking werden Software- und Hardware-Module durch Transitionssysteme formalisiert; gewünschte Eigenschaften mit Hilfe logischer Formalismen formal beschrieben; und mit Hilfe von Algorithmen automatisiert überprüft, ob ein Transitionssystem eine formal spezifizierte Eigenschaft besitzt. In dieser Veranstaltung werden die theoretischen Grundlagen des Model Checkings vermittelt, mit einem Fokus auf logik-basierten Spezifikationsprachen. Die Spezifikationsprachen LTL und CTL werden eingeführt, ihre Ausdrucksstärke untersucht, und die wichtigsten algorithmischen Ansätze für das Model Checking vorgestellt. | | | | | |
| Lehrformen: Vorlesung mit begleitenden Übungen | | | | | |
| Prüfungsformen: Abschlussprüfung; mündliche Prüfung (20-30min) oder schriftliche Klausur (120min) in Abhängigkeit der Teilnehmerzahl | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: bestandene mündliche oder schriftliche Prüfung | | | | | |

| |
|---|
| Verwendung des Moduls (in anderen Studiengängen): Studiengang Mathematik, Informatik, IT-Sicherheit |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Jun.-Prof. Thomas Zeume |
| Sonstige Informationen: Nicht im WS22/23! Ab SS23 im Sommersemester! Literatur: Einstiegsliteratur für diese Veranstaltung sind die Bücher: <ul style="list-style-type: none">• Baier, Christel, and Joost-Pieter Katoen. Principles of model checking. MIT press, 2008.• Clarke Jr, E. M., Grumberg, O., Kroening, D., Peled, D., und Veith, H. Model checking. MIT press.2018. |

| Groupware und Wissensmanagement | | | | | |
|---|-------------------------|---------------------------|--------------------------------|-------------------------------|-----------------------------|
| Modul-Nr: 2w | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Groupware und Wissensmanagement (260080) b) Übung integriert | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: 18 |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Mensch-Maschine-Interaktion | | | | | |
| Lernziele (learning outcomes): Insbesondere durch die Nutzung neuer Technologien sollen die Studierenden erfahren, wie „Groupware und Wissensmanagement“ in der Praxis funktioniert und Konzepte erarbeiten, wie die Technik, aber auch die dazugehörigen sozio-technischen Prozesse gestaltet werden müssen, um für die Nutzer/innen nutzbar und nützlich zu sein. Dieser Einsatz ändert sich jeweils im Vergleich zum Vorjahr, da er von der Art der Projektarbeit abhängt. | | | | | |
| Inhalt: In zunehmendem Maße kommen Computersysteme nicht als Individualplatzlösung, sondern in Netzwerken zum Einsatz. Diese unterstützen gemeinsames Arbeiten und Lernen und werden unter den Schlagworten Groupware und Wissensmanagement zusammengefasst. Bei der Gestaltung und Einführung solcher Systeme wird man sich immer auch mit Fragen der Organisations- und Kommunikationsveränderungen befassen müssen. Die Vorlesung ist in vier thematische Blöcke unterteilt, die aufeinander aufbauen: <ul style="list-style-type: none"> • Interdisziplinäre Grundlagen zum Verständnis von menschlicher Kommunikation, Kooperation und Organisation • Groupwaresysteme sowie Methoden und Werkzeuge • Mechanismen zur Kommunikationsunterstützung sowie zur Kooperation und Koordination • Social Software/Web 2.0, • Computer supported collaborative Learning • Wissensmanagement in Organisationen | | | | | |
| Lehrformen: Abwechslung von Vorlesungsinhalten, Seminarbeiträgen der Studierenden, Präsenzübungen und Erörterung der Projektbearbeitung | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|--|
| Prüfungsformen: Semesterbegleitend; eigener Seminarbeitrag (40 Minuten + 20 Minuten Diskussion) |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandener Seminarbeitrag (100% der Endnote) |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Thomas Herrmann Markus Jelonek Msc. |
| Sonstige Informationen: Literatur: 1. Schwabe, G., Streitz, N., Unland, R. (Hrsg.) (2001): CSCW Kompendium. Lehr- und Handbuch zum computergestützten kooperativen Arbeiten. Heidelberg u.a.: Springer. S. 25 – 35. 2. Haake, Jörg; Schwabe, Gerd; Wessner, Martin (Hrsg.) (2012): CSCL-Kompendium 2.0: Lehr- und Handbuch zum computerunterstützten kooperativen Lernen. München: Oldenbourg Verlag. |

| Industrielles Kundenmanagement | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|--|
| Modul-Nr: 2x | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Industrielles Kundenmanagement (139010) | | | Kontaktzeit: 3 SWS | Selbststudium: 105 h | Gruppengröße: 20 Studierende |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| <p>Lernziele (learning outcomes): Zielsetzung: Die Studierenden erwerben fachspezifische Kompetenzen in Form von Kenntnissen zu Grundlagen von an den Maschinenbau angrenzenden, relevanten Ingenieurwissenschaften, wie dem Wirtschaftsingenieurwesen und darüber hinaus Ansätze der Wirtschaftspsychologie und wirtschaftswissenschaftliche Aspekte. Die Studierenden sollen zentrale Methoden des Industriellen Kunden-Managements in den relevanten Management-Komponenten aktiv anwenden können und theoriegeleitet wissenschaftliche (Fall-)Studien bearbeiten, kritisch darstellen und präsentieren. Als generische Kompetenzen werden die Studierenden zu vernetztem Denken angeregt und in die Lage versetzt etablierte Methoden und Verfahren auszuwählen und anzuwenden und daraus kritisches Managementhandeln abzuleiten. Dies beinhaltet spezifische Kenntnisse und instrumentale Kompetenzen des industriellen Kunden-Management aus prozessualer, organisatorischer und strategischer Sicht. Die Studierenden erlangen vertiefte, interdisziplinäre instrumentale Kompetenzen und können diese situativ anwenden. Dabei lernen die Studierenden die Grundlagen, Methoden und Verfahren der wirtschaftsingenieur- und verhaltenswissenschaftlichen Ansätze in Praxis und Forschung im Bereich des IKM kennen. Dies beinhaltet die Analyse, Bewertung und Gestaltung der Aufbau- und Ablauforganisation und der Geschäftsmodelle unterschiedlicher Organisationen sowie wesentliche Elemente der kundenorientierten Strategieentwicklung im Kontext einer digitalen Transformation der beruflichen und privaten Lebenswelt.</p> | | | | | |
| <p>Inhalt: In den Veranstaltungen zum Industriellen Kunden-Management werden Theorien, Modelle und Ansätze dargestellt und anhand von Fallstudien und Business-Episoden vertieft und reflektiert. Die Veranstaltung orientiert sich an der zyklischen Bearbeitung von 12 wesentlichen Management-Kompetenzfeldern zum ganzheitlichen Kunden-Management. 12 Episoden bzw. Management-Kompetenzfelder:</p> <ol style="list-style-type: none"> 1. Innovationsmanagement 2. Technologiemanagement 3. FundE Management 4. Prozessmanagement | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

5. Qualitätsmanagement
6. Produktmanagement
7. Marketingmanagement
8. Vertriebsmanagement
9. Servicemanagement
10. Projektmanagement
11. Change Management
12. Kundenmanagement

Arbeitsaufwände: - Präsenzzeit: 45 h Präsenzstudium - Vor und Nachbereitung (einschl. Prüfung): 105 h Eigenstudium

Lehrformen:

Unterricht in Seminarform, Gruppenarbeiten, Bearbeitung von Fallbeispielen

Prüfungsformen:

Abschlussprüfung; Gruppenarbeit mit Präsentation zur Erlangung von Bonuspunkten (freiwillig); Mündliche Abschlussprüfung in Kleingruppen von 3-4 Personen und einer Dauer von 20 Minuten pro Prüfling

Voraussetzungen für die Vergabe von Kreditpunkten:

Teilnahme und bestehen der mündlichen Prüfung

Verwendung des Moduls (in anderen Studiengängen):

M.Sc. Maschinenbau, über den Optionalbereich auch in anderen Studiengängen

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. phil. Joachim Zülch

Dr.-Ing. Matthias Bartels

Sonstige Informationen:

Das Modul eignet sich für interessierte Studierende der Master-Phase im Optionalbereich und erfordert keine speziellen Vorkenntnisse. Weitere Informationen sowie die Unterlagen zu Vorlesungen und Übungen werden über Moodle zur Verfügung gestellt.

| Product Lifecycle Management | | | | | |
|--|-------------------------|---------------------------|--|--------------------------------|----------------------------------|
| Modul-Nr: 2y | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Product Lifecycle Management (138577) | | | Kontaktzeit: V: 2 SWS; Ü: 2 | Selbststudium: 120 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| <p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verfügen Studierende über ein breites, integriertes Wissen über die Herausforderungen des ganzheitlichen Informationsmanagements im Produktlebenszyklus und die resultierenden Anforderungen an Softwaresysteme zur Unterstützung von PLM. • kennen und verstehen Studierende die Teilprozesse des Produktlebenszyklus, die Methoden des Product Lifecycle Management (PLM) sowie die wissenschaftlichen Grundlagen der zugrundeliegenden IT-Systeme. Indem sie praktische Beispiele und Aufgaben mit entsprechender Anwendungssoftware bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit PLM Software auf konkrete und praxisorientierte PLM Problemstellungen übertragen. • haben Studierende ein umfassendes Verständnis vom Zusammenwirken der PLM Softwaresysteme und können kritisch die Eignung von Methoden zum Objektmanagement, zum Produktstruktur- und Konfigurationsmanagement sowie zum Projekt- und Prozessmanagement differenzieren und beurteilen. • können Studierende prozessorientiert an PLM Aufgabenstellungen herangehen, diese reflektieren und bewerten sowie selbstgesteuert verfolgen. • können Studierende kooperativ PLM-Aufgabenstellungen in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren. | | | | | |
| <p>Inhalt: Die Veranstaltung vermittelt Methoden und Werkzeuge zum Product Lifecycle Management (PLM), insbesondere das dazu erforderliche Grundlagenwissen und die relevanten methodischen Aspekte von Produktinnovationsprozessen. Schwerpunkte bilden dabei die verschiedenen PLM-Funktionen entsprechender Softwaresysteme (z.B. Teile-, Dokumenten- und Produktstrukturmanagement, Klassifizierung, Konfigurationsmanagement, Projekt- und Prozessmanagement) Weiterhin werden allgemeine Methoden zur Organisation und Handhabung von Produktdaten und Benutzerinformationen sowie Methoden des Collaborative Engineerings und die Vorgehensweise bei der PLM-Einführung vermittelt.</p> | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| Lehrformen: Vorlesungen, Übungen, Gruppenarbeiten, Moodle Unterstützung |
| Prüfungsformen: Abschlussprüfung; Schriftliche Klausur mit Fragen und Aufgaben zum Stoff der Vorlesung und der Übung - Dauer: 90 Minuten, Anteil der Modulnote: 100% |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulklausur Prüfungsvorleistung: Im Rahmen der Übung werden 5 Projektaufgaben gestellt, die bearbeitet und abgegeben werden müssen. Für die Zulassung zur Modulklausur müssen mind. 4 Projektaufgaben positiv bewertet sein. |
| Verwendung des Moduls (in anderen Studiengängen): M.Sc. Maschinenbau |
| Stellenwert der Note für die Endnote: 6 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Detlef Gerhard |

| Managing Digital Platform Ecosystems | | | | | |
|---|-------------------------|---------------------------|--------------------------------|--------------------------------|----------------------------------|
| Modul-Nr: 2z | Credits: 7 CP | Workload: 210 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Managing Digital Platform Ecosystems | | | Kontaktzeit: 4 SWS | Selbststudium: 150 h | Gruppengröße: beliebig |
| Sprache: Englisch | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Lernziele (learning outcomes): After completion of this module, students will understand the basic concepts of platforms as two-sided markets, model platform ecosystems, understand the design and governance of digital platforms from the perspective of the platform owner and analyze existing platform ecosystems. | | | | | |
| Inhalt: In this module, students will learn about digital platform ecosystems. Digital platforms are at the heart of business model change in digital transformation. Companies like Apple, Facebook, Uber and Google, but also Microsoft, SAP and Salesforce develop ecosystems, where partners develop a plethora of applications to match customer needs on their digital platform. Digital platform ecosystems will be analyzed from both technological design as well as the market perspective. The course addresses the managerial challenges in building and scaling platform-based business models and governing the ecosystem surrounding the digital platform. In addition to participating in the lecture, students will conduct a group case work assignment on digital platform ecosystems. | | | | | |
| Lehrformen: Vorlesung | | | | | |
| Prüfungsformen: Semesterbegleitend; Module examination, consisting of an oral exam or a graded written exam (60 minutes) (counts for 60 percent of the grading), both of them in combination with the preparation and presentation of a case (counts for 40 percent of the grading) (mode will be announced in time). | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: erfolgreich bestandene Modulprüfung | | | | | |
| Verwendung des Moduls (in anderen Studiengängen): Wahlmodul in the TU Dortmund M.Sc. Programme der Wirtschaftswissenschaften, WiWi für ein Lehramt am Berufskolleg (Modellversuch), Wirtschaftsmathematik, Wirtschaftsingenieurwesen, Logistik, Mathematik und Statistik | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

Stellenwert der Note für die Endnote: 7 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

JProf.Dr. Manuel Wiesche

JProf.Dr. Manuel Wiesche (TU Dortmund)

| Anwendungen der Computerlinguistik | | | | | |
|---|---|---|---|--|-----------------------------|
| Modul-Nr: 2aa | Credits: 10 CP oder 12 CP CP | Workload: 300 h bzw. 360 h h | Semester: 1. bis 4. . Sem. | Turnus: Jedes Semes- ter | Dauer: 1 Semester |
| Lehrveranstaltungen: Semiar | | | Kontaktzeit: 2 SWS | Selbst- studium: 270 h bzw. 330 h | Gruppengröße: 15 |
| Sprache: Deutsch oder Englisch | | | | | |
| <p>Teilnahmevoraussetzungen: Grundlegendes Wissen über die zentralen Konzepte, Modelle und Methoden der Computerlinguistik (ggf. sollten vorab die Module „Einführung in die Linguistik“ und „Methoden der Computerlinguistik“ als Freies Wahlmodul absolviert werden)</p> <p>Empfohlene Vorkenntnisse: Durcharbeiten von Standard-Lehrbüchern der Computerlinguistik, z. B. von: Jurafsky, D., Martin, J. H. (2008). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Prentice-Hall.</p> | | | | | |
| <p>Lernziele (learning outcomes): 10 CP-Variante: Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über vertieftes Wissen und Fertigkeiten in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik.</p> <ul style="list-style-type: none"> • Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten. • Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können Sie zu Ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen. <p>12 CP-Variante: Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über fundiertes Wissen zum aktuellen Forschungsstand in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik. Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung und kennen die jeweiligen Ansätze und Methoden. Basierend auf diesen Kenntnissen können Sie einen eigenen Ansatz entwickeln und diesen in einer Implementation umsetzen sowie evaluieren. Sie können zu ihrem eigenen Ansatz eine Publikation in Form eines wissenschaftlichen Aufsatzes verfassen.</p> | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|--|
| <p>Inhalt: 10 CP-Variante: In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Themenbereich werden Sie eigenständig mit computerlinguistischer Fachliteratur arbeiten, diese einordnen und bewerten und ausgewählte Ansätze nachimplementieren. 12 CP-Variante: In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Teilgebiet werden Sie eigenständig mit computerlinguistischer Fachliteratur arbeiten und eigene Ansätze entwickeln und nachimplementieren.</p> |
| <p>Lehrformen: Seminaristischer Unterricht mit Gruppenarbeiten, mündlichen Präsentationen von Forschungsarbeiten. Die Betreuung bei Selbststudiumanteilen von 150h+ pro Lehrveranstaltung erfolgt nach individueller Absprache. Im Regelfall erfolgen individuelle Sitzungen (Zoom oder Präsenz) im wöchentlichen oder 14-tägigen Wechsel.</p> |
| <p>Prüfungsformen: Implementation und Research Thesis</p> |
| <p>Voraussetzungen für die Vergabe von Kreditpunkten: Erfolgreicher Abschluss der Modulprüfung (7,0 bzw. 9 CP CP) sowie bestandene Studienleistung (3,0 CP) im Seminar.</p> |
| <p>Verwendung des Moduls (in anderen Studiengängen): Ist als Modul „Mastermodul Computerlinguistik 1 bzw. 2“ Bestandteil im Studiengang Computerlinguistik, Psycho- und Theoretische Linguistik</p> |
| <p>Stellenwert der Note für die Endnote: 10/105 bzw. 12/105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)</p> |
| <p>Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Stefanie Dipper Prof. Dr. Stefanie Dipper, Prof. Dr. Ralf Klabunde.</p> |
| <p>Sonstige Informationen: Eine Anmeldung zur Veranstaltung ist über eCampus jeweils bis zum 01.10. bzw. 01.04. vorzunehmen.</p> |

| Bioinformatics for Proteomics I | | | | | |
|---|-------------------------|---------------------------|--|--------------------------------|--|
| Modul-Nr: 2ab | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Bioinformatics for Proteomics I (201911) b) Übung | | | Kontaktzeit: V: 2 SWS; Ü: 1 | Selbststudium: 105 h | Gruppengröße: 3 bis 20 Studierende |
| Sprache: English | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Recommended prior knowledge: English and basic programming skills. | | | | | |
| <p>Lernziele (learning outcomes): Discipline-specific competences: After the successful completion of this module:</p> <ul style="list-style-type: none"> • the students have become familiar with basic knowledge of protein biochemistry, • they are able to explain the principles of mass spectrometry as the key technology of proteomics, • they are able to explain the current methods of bioinformatics for proteomics that are used for the analysis of raw data (i.e., mass spectra) in order to identify and quantify peptides or proteins, respectively, • they understand the underlying algorithmic and statistical concepts of these methods, • they are able to use proteomics-specific software and the workflow engine KNIME, • they are able to design and program own solution strategies • and they are able to apply the discussed software tools and methods to real data and problems. <p>Interdisciplinary/generic competences:</p> <ul style="list-style-type: none"> • instrumental competences: <ul style="list-style-type: none"> – Intensive usage of the learning platform Moodle • systemic competences: <ul style="list-style-type: none"> – Independent learning and working – Teamwork and ability to work in a team | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- **communicative competences:**

- Presentation of own work and results
- Communication of bioinformatics-specific technical terms
- Rhetoric and linguistic competence (English)

Inhalt:

- Basics of protein biochemistry
- Properties of amino acids
- Basics of mass spectrometry
- Raw data processing
- Protein databases
- Tryptic and in silico digest of proteins
- Principles of spectra identification search engines and scores
- Limitation of the false discovery rate using the target-decoy-approach
- PSM-specific score-correction (Percolator)
- Protein inference
- Protein quantification
- Preprocessing of quantitative data
- Quality control
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks (Tutorial)
- Workflow engine KNIME (Tutorial)

Lehrformen:

Lecture: slide-based lecture. Tutorial: Solution of small practical exercises using real example data as homework, programming tasks, group work, live-presentation of code and software and seminar-like form of teaching.

Prüfungsformen:

Abschlussprüfung; Oral examination or written test.

Voraussetzungen für die Vergabe von Kreditpunkten:

Passed oral or written examination.

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

KAPITEL 2. ANWENDUNGSBEREICH

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

PD Dr. Martin Eisenacher

Sonstige Informationen:

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

| Bioinformatics for Proteomics II | | | | | |
|--|-------------------------|---------------------------|--|--------------------------------|--|
| Modul-Nr: 2ac | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Bioinformatics for Proteomics II (201911) b) Übung | | | Kontaktzeit: V: 2 SWS; Ü: 1 | Selbststudium: 105 h | Gruppengröße: 3 bis 20 Studierende |
| Sprache: English | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Recommended prior knowledge: English, basic programming skills and lecture/tutorials “Bioinformatics for Proteomics I” in the winter term (recommended, not required). | | | | | |
| <p>Lernziele (learning outcomes): Discipline-specific competences: After the successful completion of this module</p> <ul style="list-style-type: none"> • the students have become familiar with the most important knowledge from the lecture “Bioinformatics for Proteomics I” as a brief recapitulation, • they have become familiar with the principles of advanced methods used in bioinformatics for proteomics, • they are able to explain and use advanced methods that currently are employed to analyze raw data (i.e., mass spectra) and results (i.e., peptide/protein identification and quantification results) and to interpret them biologically, • they understand the underlying algorithmic and statistical concepts of these methods, • they are able to use proteomics-specific software and the workflow engine KNIME, • they are able to design and program own solution strategies • and they are able to apply the discussed software tools and methods to real data and problems. <p>Interdisciplinary/generic competences:</p> <ul style="list-style-type: none"> • instrumental competences: <ul style="list-style-type: none"> – Intensive usage of the learning platform Moodle • systemic competences: <ul style="list-style-type: none"> – Independent learning and working | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

– Teamwork and ability to work in a team

• **communicative competences:**

- Presentation of own work and results
- Communication of bioinformatics-specific technical terms
- Rhetoric and linguistic competence (English)

Inhalt:

- Brief recapitulation of “Bioinformatics for Proteomics I”
- Computational comparison of protein lists
- Statistics for the comparison of experimental groups
- Machine learning-based biomarker discovery (supervised and unsupervised methods)
- Enrichment analysis
- Network analysis
- Single / multiple / parallel reaction monitoring (SRM / MRM / PRM)
- Data independent acquisition (DIA)
- Algorithms for de novo sequencing of peptides
- Open searches
- Dark matter of proteomics
- Proteoforms
- Metaproteomics and proteogenomics
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks (Tutorial)
- Workflow engine KNIME (Tutorial)

Lehrformen:

Lecture: slide-based lecture. Tutorial: Solution of small practical exercises using real example data as homework, programming tasks, group work, live-presentation of code and software and seminar-like form of teaching.

Prüfungsformen:

Abschlussprüfung; Oral examination or written test

Voraussetzungen für die Vergabe von Kreditpunkten:

Passed oral or written examination.

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

KAPITEL 2. ANWENDUNGSBEREICH

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

PD Dr. Martin Eisenacher

Sonstige Informationen:

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

| Bioimage Informatics | | | | | |
|---|-------------------------|---------------------------|--------------------------------|-------------------------------|----------------------------------|
| Modul-Nr: 2ad | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes WiSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Vorlesung Bioimage Informatics (190800) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: beliebig |
| Sprache: Deutsch | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Grundlegende Programmierkenntnisse | | | | | |
| Lernziele (learning outcomes): Die Studierenden sollen elementare und aktuelle Techniken der Bildverarbeitung kennenlernen, die zur Analyse mikroskopischer Bilddaten verwendet werden. Sie sollen die dahinterliegenden algorithmischen, mathematischen und statistischen Ideen verstehen, und lernen in der Praxis, wie diese Methoden auf reale Daten und Fragestellungen angewendet werden. | | | | | |
| Inhalt: Die Analyse von mikroskopischen Bilddaten mit Methoden der Bildverarbeitung ist in den vergangenen Jahren ein wichtiges Thema in vielen Anwendungen in den Lebenswissenschaften geworden. In der Vorlesung werden grundlegende Konzepte der Verarbeitung mikroskopischer Bilddaten und deren Anwendungen behandelt. Die Vorlesung gliedert sich wie folgt: | | | | | |
| <ul style="list-style-type: none"> • Grundlagen der Mikroskopie: Durchlicht- und Konfokalmikroskopie; Fluoreszenzmikroskopie; Mikroskopie in der Histopathologie; Mikroskopie in der Zellbiologie; Mikroskopie in der Neurobiologie • Morphologische Bildanalyse • Texturelle Bildanalyse • Kolokalisations-Verfahren • Algorithmen zur Bildregistrierung • Algorithmen zum Verfolgen von Bewegungsmustern von Zellen (“Cell Tracking”) • Verfahren zur Rekonstruktion von Neuronen (“Neuron Tracing”) • Deep Learning Methoden zur mikroskopischen Bilddatenanalyse • Methoden der Analyse von markerfreien Mikroskopie-Daten • Methoden zur überwachten und unüberwachten Segmentierung von mikroskopischen Bildern | | | | | |
| Lehrformen: | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

| |
|---|
| b) Übung |
| Prüfungsformen: Abschlussprüfung; mündliche Prüfung |
| Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene mündliche Prüfung |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu |
| Stellenwert der Note für die Endnote: 5 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: PD Dr. Axel Mosig |
| Sonstige Informationen: Weitere Informationen unter http://www.bioinf.rub.de/msc/ |

| Master-Praktikum Big Data in der Bioinformatik | | | | | |
|---|-------------------------|---------------------------|--------------------------------|----------------------------------|---|
| Modul-Nr: 2ae | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: Jedes Semester | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Praktikum (202621) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: 1-3 Studierende |
| Sprache: Deutsch oder Englisch | | | | | |
| Teilnahmevoraussetzungen: praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab) | | | | | |
| Empfohlene Vorkenntnisse: Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik | | | | | |
| Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls | | | | | |
| <ul style="list-style-type: none"> • können Studierende praktischen Herausforderungen bei der Entwicklung von Bioinformatik-Anwendungen begegnen, • haben die Studierenden Programmier-Bibliotheken aus dem Bereich Bioinformatik kennen gelernt • haben Studierenden die Verwendung von Workflow-Systemen eingeübt • haben Studierende die Mechanismen von Code- und Programm-Dokumentation eingeübt • haben Studierende Verfahren zur Bereitstellung eigener Libraries (z. B. R-Pakete, .jar-Files) angewendet | | | | | |
| Vermittelte Kompetenzen: | | | | | |
| Fachspezifische Kompetenzen: | | | | | |
| <ul style="list-style-type: none"> • Programmieren von Lösungen für bioinformatische Anwendungen • Umgang mit Standard-Werkzeugen / -Formaten und Programmier-Bibliotheken der Bioinformatik • Umgang mit größeren Datenmengen | | | | | |
| Generische Kompetenzen: | | | | | |
| <ul style="list-style-type: none"> • instrumentale Kompetenzen | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- Projekt- und Zeitmanagement mit digitalen Tools
- Wissenschaftlich gegliederte Dokumentation

- **systemische Kompetenzen**

- Teamarbeit und Teamfähigkeit
- Selbständiges Lernen und Arbeiten

- **kommunikative Kompetenzen**

- Präsentation wissenschaftlicher Ergebnisse
- Vermittlung der in der Bioinformatik angewandten englischen Fachbegriffe
- Rhetorik und sprachliche Kompetenz (deutsch / englisch)

Inhalt:

Die Bioinformatik wendet naturgemäß Informatik-Methoden auf die Daten eines lebenswissenschaftlichen Anwendungsfaches an, stellt also per se angewandte Digitalisierung dar. Das Praktikum vermittelt Grundlagen der Programmierung mit Bezug zu lebenswissenschaftlichen Anwendungen mit großen Datenmengen. Dies geschieht anhand aktueller Beispiele aus den Themengebieten Bildverarbeitung und Sequenzanalyse. Nach einer kurzen Einführung (Präsenztreffen) in Programmierung und Entwicklungsumgebungen (z. B. Java, R, C++, eclipse, RStudio, Python, Matlab) werden praktische Programmieraufgaben ausgegeben und im Laufe des Semesters bearbeitet, ggfls. mit weiteren Präsenztreffen zur Diskussion des Fortschritts.

Lehrformen:

Einführung als seminaristischer Unterricht, Bearbeitung der praktischen Aufgabe selbständig oder als Gruppenarbeit

Prüfungsformen:

Semesterbegleitend; Praktische Prüfung: Protokoll max. 10 Seiten (wenn Abgabe bis zwei Wochen nach Praktikumsende, dann Korrekturrunde vor Benotung), Dokumentation der praktischen Aufgabe

Voraussetzungen für die Vergabe von Kreditpunkten:

Protokoll (siehe Prüfungsformen), Bearbeitung / Dokumentation der praktischen Aufgabe, Teilnahme an den Präsenztreffen, Abschlusspräsentation (15 min. + 5 min. Diskussion)

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:**Prof. Dr. Axel Mosig**

Prof. Dr. Axel Mosig, Prof. Dr. Sven Rahmann (Universität Duisburg/Essen und TU Dortmund)

Sonstige Informationen:

KAPITEL 2. ANWENDUNGSBEREICH

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

| Master-Praktikum Computational Proteomics | | | | | |
|--|-------------------------|---------------------------|--------------------------------|-------------------------------|---|
| Modul-Nr: 2af | Credits: 5 CP | Workload: 150 h | Semester: 1.-3. Sem. | Turnus: jedes SoSe | Dauer: 1 Semester |
| Lehrveranstaltungen: a) Praktikum (202621) | | | Kontaktzeit: 4 SWS | Selbststudium: 90 h | Gruppengröße: 1-3 Studierende |
| Sprache: Deutsch oder Englisch | | | | | |
| Teilnahmevoraussetzungen: Practical programming experience in one or more languages (e. g. Java, R, Python, Matlab) Empfohlene Vorkenntnisse: Modules from the focal point Bioinformatics (e.g. lecture “Bioinformatics for Proteomics I”) | | | | | |
| <p>Lernziele (learning outcomes): After successful completion of the module students:</p> <ul style="list-style-type: none"> • can take on practical challenges in the development of bioinformatics applications, • the students have got to know programming libraries from the field of bioinformatics • have practiced the use of workflow systems • have learned the mechanisms of code and program documentation • have practiced procedures used to provide own libraries (e.g. R packages, .jar files) <p>Discipline-specific competences:</p> <ul style="list-style-type: none"> • Programming of solutions for bioinformatics applications • Handling of standard tools / formats and programming libraries of bioinformatics • Dealing with large amounts of data <p>Generic competences:</p> <ul style="list-style-type: none"> • Instrumental competences <ul style="list-style-type: none"> – Project and time management with digital tools – Scientifically structured documentation • Systemic competences <ul style="list-style-type: none"> – Teamwork and ability to work in a team – Independent learning and working • Communicative skills | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

- Presentation of scientific results
- Using English technical terms used in Bioinformatics
- Rhetoric and linguistic competence (German / English)

Inhalt:

Bioinformatics naturally applies computer science methods to the data of a life science application area, i. e. it represents applied digitization per se. The internship teaches the basics of programming / development with relations life science applications utilizing data of high-throughput mass spectrometry / Proteomics. After a short introduction (presence meeting) in programming and development environments (e.g. Java, R, C++, eclipse, RStudio, Python, Matlab, KNIME), practical programming tasks will be enrolled and edited during the semester, if necessary with further face-to-face meetings to discuss progress.

Lehrformen:

Introductory weeks, independent working on the practical task alone or as group work

Prüfungsformen:

Semesterbegleitend; Written protocol with the practical's topics and the work's results (max. 10 pages), documentation of the practical task, final slide presentation (15 min. + 5 min. discussion)

Voraussetzungen für die Vergabe von Kreditpunkten:

Written protocol (see "Prüfungsformen"), processing / documentation of the practical task, participation in the face-to-face meetings, final presentation

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 5 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

PD Dr. Martin Eisenacher

KAPITEL 2. ANWENDUNGSBEREICH

| Fachwissenschaftliche Vertiefung | | | | | | |
|--|-------------------------|---------------------------|---|--------------------------------|-----------------------------------|--|
| Modul-Nr: 3 | Credits: 6 CP | Workload: 180 h | Semester: 1.-3. Sem.. Sem. | Turnus: immer | Dauer: 1-2 Semester | |
| Lehrveranstaltungen: a) Vertiefungsseminare nach Wahl | | | Kontaktzeit: 4 SWS | Selbststudium: 120 h | Gruppengröße: a) ca. 20 | |
| Sprache: Deutsch oder Englisch | | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | | |
| Lernziele (learning outcomes): a) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verfügen Studierende über aktuelle fachwissenschaftliche Kenntnisse aus 2 verschiedenen Vertiefungsrichtungen • können die Teilnehmer Lehrinhalte und Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln • können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen • sind die Studierenden in der Lage, die Ergebnisse der eigenen Arbeit schriftlich zu dokumentieren | | | | | | |
| Inhalt: Es werden Vertiefungsseminare zu mehreren relevanten Themen angeboten, wie beispielsweise zu Deep Learning, zu Bioinformatik oder zu Ingenieurinformatik. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen in der Regel 2 verschiedene Seminarethemen aus. Diese sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation sowie eine schriftliche Ausarbeitung. Nähere Informationen zu den jeweiligen Seminaren sind dem Vorlesungsverzeichnis zu entnehmen. Anmerkung: Die meisten Seminare werden mit 3 CP kreditiert. Hier müssen 2 verschiedene Seminare absolviert werden (z.B. Ingenieurinformatik und Knowledge Graphs). Das Seminar „Student Conference: Research Project“ hat einen Workload von 6 CP und ist auch als Einzelseminar ausreichend, um das Modul abzuschließen. | | | | | | |
| Lehrformen: Seminarvorträge und Feedbackrunden | | | | | | |
| Prüfungsformen: Semesterbegleitend; 1-2 Seminarvorträge zu unterschiedlichen Vertiefungsbereichen und evtl. jeweils eine schriftliche Ausarbeitung | | | | | | |

KAPITEL 2. ANWENDUNGSBEREICH

Voraussetzungen für die Vergabe von Kreditpunkten:

Die meisten Seminare werden mit 3 CP kreditiert. In diesem Fall errechnet sich die Modulnote als arithmetisches Mittel aus beiden Seminarnoten. Bei Seminaren mit 6 CP ist die jeweilige Seminarnote auch die Modulnote. Die Gesamtnote muss mindestens "ausreichend" sein. Um die Lernziele zu erreichen, besteht im Seminare Anwesenheitspflicht an mindestens 9 von 10 Terminen. Mehrfaches Fehlen muss durch ein ärztliches Attest entschuldigt werden. Die Anwesenheit beim ersten Termin ist obligatorisch, da zu diesem Termin die Themen verteilt werden. Ein Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 6 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:**Studiendekan Angewandte Informatik**

Je nach Wahl der Seminare

Freie Wahlmodule

| Freie Wahlmodule | | | | | |
|--|----------------------------------|---------------------------|---|---|----------------------------------|
| Modul-Nr: 4 | Credits: Min. 15 CP | Workload: 450 h | Semester: 1.-6.. Sem. | Turnus: Jedes Semes- ter | Dauer: 1 Semester |
| Lehrveranstaltungen: Frei wählbar | | | Kontaktzeit: Je nach Lehrveran- staltungswahl | Selbst- studium: Je nach Lehrveran- staltungswahl | Gruppengröße: beliebig |
| Sprache: Je nach Lehrveranstaltung | | | | | |
| Teilnahmevoraussetzungen: Keine | | | | | |
| Empfohlene Vorkenntnisse: Abhängig von der gewählten Veranstaltung | | | | | |
| Lernziele (learning outcomes): Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern. | | | | | |
| Vermittelte Kompetenzen: Kernkompetenzen: <ul style="list-style-type: none">• Fächerübergreifendes Denken und Arbeiten | | | | | |
| Inhalt: Die freien Wahlfächer erweitern die Soft Skills. Z.B. wird die englische Fachsprache verbessert, in die Grundlagen der Rechtswissenschaften eingeführt oder Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit eine Auswahl entsprechend der eigenen Interessen zu treffen. | | | | | |
| Lehrformen: Frei wählbar | | | | | |
| Prüfungsformen: Abhängig von der gewählten Veranstaltung | | | | | |
| Voraussetzungen für die Vergabe von Kreditpunkten: | | | | | |

KAPITEL 3. FREIE WAHLMODULE

| |
|---|
| Bestandene Modulabschlussprüfung |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu |
| Stellenwert der Note für die Endnote: unbenotet (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Studiendekan Angewandte Informatik Dozenten der RUB |
| Sonstige Informationen: Aus der Fakultät für Wirtschaftswissenschaft sind nur die folgenden Veranstaltungen als freie Wahlfächer für die AI geöffnet: <ul style="list-style-type: none">• Finanzierung und Investition• Grundlagen der Businessplanerstellung• Grundlagen der Existenzgründung• Grundlagen der Makroökonomie• Grundlagen der Mikroökonomie• Grundlagen des Wirtschaftsrechts• Jahresabschluss• Märkte und Unternehmungen• Marktorientierte Unternehmensführung <p>Der Besuch weiterer Veranstaltungen aus der Fakultät für Wirtschaftswissenschaft ist nur nach Absprache mit dem Dozierenden und dem Prüfungsamt AI möglich.</p> |

Studienprojekt

| Master-Studienprojekt | | | | | |
|--|--------------------------|---------------------------|------------------------------|----------------------------------|-----------------------------|
| Modul-Nr: 5 | Credits: 10 CP | Workload: 300 h | Semester: 3. Sem. | Turnus: Jedes Semester | Dauer: 1 Semester |
| Lehrveranstaltungen: Studienprojekt nach Wahl | | | Kontaktzeit: 2 SWS | Selbststudium: 270 h | Gruppengröße: 2-6 |
| Sprache: Deutsch oder Englisch je nach Projektwahl | | | | | |
| Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Bitte entnehmen Sie die projektspezifischen Vorkenntnisse dem jeweiligen Studienprojekt. | | | | | |
| Lernziele (learning outcomes): Ziele des Studienprojekts sind: <ul style="list-style-type: none"> • Die Anwendung des erlernten Fachwissens. • Der Erwerb zusätzlicher Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung. • Die Schulung hinsichtlich der Erarbeitung eigener Lösungsstrategien. • Die Schulung hinsichtlich Arbeitsteilung und Zusammenarbeit im Team (Teamfähigkeit und Projektorganisation). • Erwerb weiterer Kompetenz im Hinblick auf die Dokumentation und die Präsentation von Ergebnissen. | | | | | |
| Inhalt: Im Rahmen des Studienprojekts soll eine Aufgabe aus Bereichen der Angewandten Informatik in Teamarbeit unter Anleitung eines Betreuers gelöst werden. Organisation und Ablauf: Das Studienprojekt muss vor Beginn beim Prüfungsamt angemeldet werden. Bitte beachten Sie, die hierfür festgelegten Fristen unter Sonstiges. In begründeten Fällen (Industriekooperation etc.) können Studienprojekte auch außerhalb dieses Verfahrens | | | | | |

KAPITEL 4. STUDIENPROJEKT

durchgeführt werden, wenn Sie durch den Prüfungsausschuss AI beantragt und genehmigt wurden.

Durchführungszeitraum: Studienprojekte starten mit Beginn der Vorlesungszeit und enden in der Regel eine Woche nach Vorlesungsende durch die Abgabe einer Abschlussdokumentation. Die termingerechte Abgabe und positive Bewertung durch den Dozenten sind Voraussetzung für die Vergabe der Leistungspunkte.

Lehrformen:

Projektarbeit

Prüfungsformen:

Semesterbegleitend; Projektarbeit

Voraussetzungen für die Vergabe von Kreditpunkten:

Aktive Teilnahme am Projekt, fristgerechte Abgabe des Abschlussbereichs und Bewertung des Abschlussberichts mit min. ausreichend. Hinweis: Jeder Abschlussbericht muss im Umfang von einer DinA4 Seite eine Reflexion über das methodische Vorgehen enthalten.

Verwendung des Moduls (in anderen Studiengängen):

trifft nicht zu

Stellenwert der Note für die Endnote: 10 / 105

(Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet)

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann

Dozenten der RUB

Masterarbeit und Kolloquium

| | | | | | |
|---|--------------------------|---------------------------|------------------------------|----------------------------------|-----------------------------|
| Masterarbeit | | | | | |
| Modul-Nr: 6 | Credits: 30 CP | Workload: 900 h | Semester: 4. Sem. | Turnus: Jedes Semester | Dauer: 1 Semester |
| Lehrveranstaltungen: n/a | | | Kontaktzeit: 0 SWS | Selbststudium: 900 h | Gruppengröße: 1 |
| Sprache: Deutsch oder Englisch | | | | | |
| Teilnahmevoraussetzungen: erfolgreich abgeschlossene Module im Umfang von mindestens 60 CP Empfohlene Vorkenntnisse: Fachspezifisch von der Themenstellung abhängig. | | | | | |
| Lernziele (learning outcomes): Nach erfolgreichem Abschluss des Moduls: <ul style="list-style-type: none"> • können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate • können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen, anwenden und weiterentwickeln, um ein konkretes Problem zu lösen • können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren • können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen | | | | | |
| Inhalt: Es soll eine anspruchsvolle Fragestellung der Angewandten Informatik bearbeitet und dokumentiert werden. Im Anschluss an die Bearbeitung der Masterarbeit werden die Ergebnisse in Form eines Kolloquiumsvortrags mit anschließender Diskussion präsentiert. | | | | | |
| Lehrformen: Projektarbeit | | | | | |

KAPITEL 5. MASTERARBEIT UND KOLLOQUIUM

| |
|--|
| Prüfungsformen: Abschlussarbeit (90% der Note) + Kolloquium (10% der Note) |
| Voraussetzungen für die Vergabe von Kreditpunkten: erfolgreiche schriftliche Masterarbeit + Kolloquiumsvortrag |
| Verwendung des Moduls (in anderen Studiengängen): trifft nicht zu |
| Stellenwert der Note für die Endnote: 30 / 105 (Im Studiengang werden Module im Umfang von 105 CP benotet und 15 CP nicht benotet) |
| Modulbeauftragte/r und hauptamtlich Lehrende: Studiendekan der Angewandten Informatik Dozierende der RUB |
| Sonstige Informationen: Im Normalfall sucht sich jede bzw. jeder Studierende nach eigenem Interesse und Neigung einen Lehrstuhl aus, an dem sie bzw. er die Masterarbeit schreiben möchte. Die meisten Lehrstühle veröffentlichen ihre angebotenen Themen. Oft werden Themen aber auch erst nach Absprache mit demn Studierenden gestellt, wobei Letzterer ein Vorschlagsrecht hat. Die Anmeldung für die Masterarbeit erfolgt beim Prüfungsamt Angewandte Informatik. Für die Anmeldung ist das persönliche Erscheinen sowie die Vorlage des aktuellen Studentenausweises bzw. der Immatrikulationsbescheinigung notwendig. |