

M.Sc.

Ingenieurinformatik

(Winter 2025/26)

- [Modulhandbuch](#)
- [☐☐ Das neue Modulhandbuch](#)
- [☐☐ Orientierung](#)
 - [Über den Studiengang](#)
 - [Orientierung am Regelstudienplan](#)
 - [Hinweise zu Modulen im Bereich: Ingenieurwissenschaften](#)
- [☐☐ Neuigkeiten](#)
 - [Nutzung von generativer KI für Abschlussarbeiten und Prüfungs\(vor\)leistungen](#)
 - [Fachschaftsrat der Fakultät für Informatik \(Studierendenvertretung\)](#)
- [⚠ Regelungen des Prüfungsausschusses](#)
- [Module im Bereich: Informatik](#)
 - [AMS Lab Projects](#)
 - [Advanced Topics in Deep Learning \(Master\)](#)
 - [Advanced Topics in Networking](#)
 - [Applied Discrete Modelling](#)
 - [Ausgewählte Algorithmen der Computergraphik](#)
 - [Clean Code Development](#)

- [Computer-Assisted Surgery](#)
- [Data Mining II - Advanced Topics in Data Mining](#)
- [Data Science with Python](#)

- [Module im Bereich: Ingenieurinformatik](#)
 - [AMS Lab Projects](#)

- [Module im Bereich: Schlüssel- und Methodenkompetenzen](#)
 - [Clean Code Development](#)

- [Masterarbeit](#)
- [Feedback](#)
- [Credits](#)

Modulhandbuch

für den Master of Science (M.Sc.)
Ingenieurinformatik

für das Wintersemester 2025/26



FAKULTÄT FÜR
INFORMATIK

der Fakultät für Informatik

gültig für die SPO 2024 und ältere SPOs*

Stand 09.10.2025

* Für ältere SPOs als die SPO 2024 gelten abweichende Regelstudienpläne, die entsprechend berücksichtigt werden müssen. Beachten Sie insbesondere abweichende Pflichtveranstaltungen und CP. Nicht zu Ihrer SPO passende Lehrveranstaltungen können ggf. nicht vollständig oder gar nicht angerechnet werden. Prüfen Sie auch die Zuordnungen aus dem

Modulhandbuch des Wintersemesters 2024/2025. Wenden Sie sich bei Inkonsistenzen oder Unklarheiten an das Prüfungsamt.

Redaktionelle Änderungen im Vergleich zum Fakultätsratsbeschluss vom 03.09.2025:

- LSF-Verlinkung korrigiert.

Das neue Modulhandbuch

Herzlich Willkommen zum neuen Modulhandbuch!

Wir hoffen, die neue Version erleichtert unseren Studierenden die Studienplanung.

Motivation

Bis zum Wintersemester 2024/25 gab es ein Modulhandbuch für alle Studiengänge, was für die Studierenden oft unübersichtlich war. Das neue Format für unsere Modulhandbücher ändert das.

Was ist neu?

Ab dem Sommersemester 2025 wird für jeden Studiengang ein Modulhandbuch für das Winter- und das Sommersemester bereitgestellt.

Darin befinden sich alle Module, die im aktuellen Semester angeboten werden und eine vorläufige Sicht auf das folgende Semester.

In jedem Modulhandbuch befinden sich zudem weitere Hinweise zur besseren Studienplanung.

Diese Änderungen sind das Ergebnis der letzten Studiengangsgespräche und sollen den Studierenden helfen, eine aktuelle, verlässliche und leicht zugängliche Übersicht ihrer Studieninhalte zu erhalten.

Was findet man darin?

Es gibt separate Buchregale (Shelves) für Bachelor- und Masterstudiengänge sowie für Sommer- und Wintersemester. In jedem Regal (Shelf) sind die spezifischen Handbücher für die einzelnen Studiengänge zu finden.

In jedem Handbuch gibt es folgende Inhalte:

1. Ein **Orientierungskapitel** mit Informationen über den Studiengang, einer Übersicht der Module anhand eines beispielhaften Regelstudienplans und Hinweisen zu Modulen aus anderen Fakultäten.
2. Ein Kapitel über **Neuigkeiten** im Studiengang oder an der Fakultät, beispielsweise zu Modulen, die im aktuellen Semester nicht angeboten werden oder zu neu eingeführten

Modulen.

3. Mehrere Kapitel, die alle **Module auflisten**, die gemäß dem Regelstudienplan einem bestimmten Modulbereich zugeordnet sind.
4. Eine Seite mit Hinweisen und **Regelungen des Prüfungsausschusses**, die für das Studium relevant sind.
5. Eine Seite mit einem Link zu einem **Feedback-Formular**, um Anregungen zu sammeln, die unsere Modulhandbücher in Zukunft weiter verbessern.

Die Fakultät wünscht ihren Studierenden viel Erfolg mit dem neuen Modulhandbuch!

□□ Orientierung

Über den Studiengang

Studienziel

Ziel ist die auf einem einschlägigen Bachelorstudiengang aufbauende wissenschaftliche Weiterqualifikation. Ein breites Angebot von Wahlpflichtfächern und Angebote in Englisch unterstützen eine weitgehende individuelle Gestaltung des Studiengangs und eine internationale Ausrichtung. Der Studiengang umfasst den Erwerb von Fähigkeiten zur wissenschaftlichen Arbeit durch Erschließen, Weiterentwickeln und Anwenden wissenschaftlicher Konzepte, Methoden und Werkzeuge im interdisziplinären Kontext, den Erwerb von Fähigkeiten zur selbstständigen, systematischen und methodischen Einarbeitung in neue Themengebiete sowie der Erwerb von Fertigkeiten im Umgang mit fachgebietsbedigten Werkzeugen.

Spätere Berufsfelder

Anspruchsvolle Tätigkeiten und Leitungsfunktionen auf den Gebieten der Konzipierung und Realisierung komplexer Systeme in allen Bereichen in denen Informationstechnologie eingesetzt wird.

Notwendige Kenntnisse / Erfahrungen / Interessen

Gute bis sehr gute Kenntnisse in den Basisfächern der Informatik und Ingenieurwissenschaften

Weitere Informationen

Hier geht es zu weiteren [Studiendokumenten](#).

Orientierung am Regelstudienplan

Die Grafik zeigt den Regelstudienplan mit Studienbeginn im Wintersemester. Verlinkt sind alle Module und Modulbereiche. Links im Winter-/Sommersemester führen unter Umständen zum Wechsel des Modulhandbuchs. Das dient einer besseren Orientierung und Studienplanung. Der Regelstudienplan nach SPO ist unter [Studiendokumente](#) zu finden.

Beispiel mit Beginn im Wintersemester

1. Semester	2. Semester	3. Semester
Informatik		Masterarbeit
Ingenieurinformatik		
Ingenieurwissenschaften		
Schlüssel- und Methodenkompetenz		

Die Module sind zu thematischen Gruppen zusammengefasst:

- Informatik
- Ingenieurinformatik
- Ingenieurwissenschaften

- Schlüssel- und Methodenkompetenz (SMK)

Hinweise zu Modulen im Bereich: Ingenieurwissenschaften

Im Bereich Ingenieurwissenschaften belegen Studierende beliebige Module aus den Fakultäten FEIT, FMB und FVST. Die Auswahl sollte sich an den individuellen Interessen der Studierenden orientieren. Es ist auch möglich, Module aus unterschiedlichen Fakultäten zu kombinieren. Dies kann sinnvoll sein, da moderne Herausforderungen in der Ingenieurinformatik oft disziplinübergreifend sind.

Ausgeschlossen sind dabei Module, die eindeutig Informatik-Themen (z.B. *Grundlagen der Informatik für Ingenieure*, *Python in Production* *System Engineering*) behandeln bzw. von FIN-Lehrenden in die Ingenieurfakultäten exportiert werden.

Abweichende Module müssen vom Studiengangsverantwortlichen genehmigt werden.

Elektrotechnik

- Regenerative Elektroenergiequellen-Systembetrachtung (Winter)
- Unkonventionelle elektrische Maschinen (Winter)
- Elektromagnetische Verträglichkeit (Sommer)
- Regelung von Drehstrommaschinen (Sommer)
- Systemintegration von Leistungselektronik (Sommer)

Module aus anderen Fakultäten sind in diesem Modulhandbuch nicht enthalten; sie können in den Modulhandbüchern der [entsprechenden Fakultäten](#) gefunden werden.

☐☐ Neuigkeiten

Nutzung von generativer KI für Abschlussarbeiten und Prüfungs(vor)leistungen

Der Fakultätsrat (Beschluss 019/25) hat folgende Regelung für die Nutzung von generativer KI für Abschlussarbeiten und Prüfungs(vor)leistungen beschlossen:

“ Die Verwendung von Inhalten, die durch künstliche Intelligenz (KI) in einer Arbeit oder einer Antwort auf eine Frage generiert wurden (einschließlich, aber nicht beschränkt auf Text, Abbildungen, Bilder und Code), muss offengelegt werden. Das verwendete KI-System ist anzugeben, und bestimmte Abschnitte der Arbeit bzw. Antwort, in denen KI-generierte Inhalte verwendet werden, sind zu kennzeichnen und mit einer kurzen Erklärung zu versehen, auf welcher Ebene das KI-System zur Generierung der Inhalte verwendet wurde. Ebenso soll der Grund für die Verwendung der Tools angegeben werden.

Die Verwendung von KI-Systemen für die Bearbeitung und Verbesserung der Grammatik ist gängige Praxis. Auch in diesem Fall ist eine Offenlegung wie oben beschrieben erforderlich, sie kann aber summarisch für die gesamte Arbeit erfolgen.

Auch bei Verwendung von generativer KI muss die wissenschaftliche Eigenleistung selbst erbracht werden.

Fachschaftsrat der Fakultät für Informatik (Studierendenvertretung)

Der Fachschaftsrat der Fakultät für Informatik (kurz: [FaRaFIN](#)) ist die gewählte Vertretung der Studierenden der Fakultät für Informatik.

Man wird ihn am ehesten durch die Organisation diverser Veranstaltungen, wie der [Einführungswoche](#), der [Vorkurse](#), Spieleabenden, Weihnachtsfeiern und Sommerfesten, kennen.

Außerdem vertreten sie die Angelegenheiten der Studierenden zur Verbesserung des Studiums und der Lehre in verschiedenen Kommissionen, wie dem Fakultätsrat, Studienkommissionen oder dem Prüfungsausschuss.

Regelungen des Prüfungsausschusses

Diese Sammlung von Regelungen fasst relevante Beschlüsse des Prüfungsausschusses der Fakultät für Informatik zusammen. Im Zweifelsfall gilt der Wortlaut der Beschlüsse des Prüfungsausschusses.

[Sie sind hier zu finden.](#)

Module im Bereich: Informatik

AMS Lab Projects

(in German: AMS Lab Projects)

Module-ID: FIN-INF-999912

Link:	LSF
Responsibility:	Prof. Dr.-Ing. Benjamin Noack
Lecturer:	Prof. Dr.-Ing. Benjamin Noack
Classes:	<i>AMS Lab Projects</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. INGINF: Ingenieurinformatik - M.Sc. WIF: Informatik - M.Sc. DKE: Applied Data Science - M.Sc. DE: Interdisziplinäres Team-Projekt - M.Sc. DE: Digital Engineering-Projekt - M.Sc. VC: Computer Science

Abbreviation AMS-Lab	Credit Points 6	Semester every	Term 2.	Duration 1 Semester	Language english	Level Master
--------------------------------	---------------------------	--------------------------	-------------------	-------------------------------	----------------------------	------------------------

Intended learning outcomes:

Students who complete the course ...

- can collaborate in teams to solve challenges in technical domains and deepen their technical expertise.
- can manage a project and define goals and milestones
- can acquire the knowledge in a technical domain to derive solutions
- can present and document their work and achievements

Content:

This module offers different team projects and individual projects on current, research-related

areas like:

- robotics
- sensor data processing
- navigation and planning
- optimization and control
- and more

Workload:

180 h of individual and team work

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	• Software and Documentation • Project report • Two Presentations	Project (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	

Media:	Literature:
	Literature will be announced in the lecture.

Comments:

Advanced Topics in Deep Learning (Master)

(in German: Advanced Topics in Deep Learning (Master))

Module-ID: FIN-INF-120496

Link:	LSF
Responsibility:	Sebastian Stober
Lecturer:	Sebastian Stober
Classes:	<i>Advanced Topics in Deep Learning</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. DKE: Learning Methods and Models for Data Science - M.Sc. DE: Methoden der Informatik - M.Sc. VC: Computer Science

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ATDL	6	jedes	ab 1.	1 Semester	english	Master

Intended learning outcomes:

Students ...

- can read and analyze current research papers
- apply deep learning to an application domain
- gain in-depth knowledge In the application domain
- learn to work efficiently in a team on scientific problems
- learn to work on a server infrastructure
- present scientific results in a written and oral form

Content:

- Current topics from deep learning research
- Create domain-specific deep learning pipelines
- The application domain will be announced beforehand

Workload:

- 30h in person
- 150h self-study

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	• Hausarbeit • Mündliche Prüfung	Vorlesung/Seminar (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
none	Introduction to Deep Learning

Media:	Literature:

Comments:

Advanced Topics in Networking

(in German: Advanced Topics in Networking)

Module-ID: FIN-INF-140014

Link:	LSF
Responsibility:	David Hausheer
Lecturer:	David Hausheer
Classes:	• Lecture <i>Advanced Topics in Networking</i> • Exercise <i>Advanced Topics in Networking</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. DE: Methoden der Informatik - M.Sc. DE: Fachliche Spezialisierung - M.Sc. VC: Computer Science

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ATN	6	Winter	ab 1.	1 Semester	english	Master

Intended learning outcomes:

Students...

- enhance their competence in the area of networking
- understand advanced networking concepts
- can explain these concepts to other participants
- are able to apply these concepts in theoretical use cases and practical network applications

Content:

The course deals with advanced topics in the area of networking:

- Overlay networks for content delivery, e.g., P2P, BitTorrent, CDNs, caching, overlay video streaming
- Distributed Hash Tables (DHT), e.g., Kademlia
- Blockchains
- Crypto currencies and Bitcoin
- Ethereum and smart contracts
- Secure network architectures, e.g., SCION
- Congestion Control, e.g., QUIC and Multipath-QUIC

Workload:

- 28h lectures
- 6h practical exercises
- 146h homework (further studies, theoretical exercises, preparation for the final exam)

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	Written exam 120 minutes	• Lecture 2 SWS • Exercise 2 SWS

Prerequisites according to examination regulations:	Recommended prerequisites:
none	Modul <i>Computernetze</i>

Media:	Literature:
	Textbooks as indicated. Slides and paper copies as necessary.

Comments:

Applied Discrete Modelling

(in German: Applied Discrete Modelling)

Module-ID: FIN-INF-120279

Link:	LSF
Responsibility:	PD Dr. Claudia Krull
Lecturer:	PD Dr. Claudia Krull
Classes:	• Lecture <i>Applied Discrete Modelling</i> • Exercise class <i>Applied Discrete Modelling</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. DKE: Learning Methods and Models for Data Science - M.Sc. DE: Fachliche Spezialisierung - M.Sc. VC: Computer Science

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
ADM	6	Winter	ab 1.	1 Semester	english	Master

Intended learning outcomes:

- Students understand the basics of Markov chains and know selected applications and solution methods
- Students understand non-Markovian stochastic processes and can model and simulate them in different ways
- Students understand hidden Markovian and non-Markovian processes
- Students know selected research topics of the research group
- Students can implement the models and methods they have learned and apply them to problems from the university's main research areas, in particular medicine and engineering

Content:

- Discrete-time and continuous-time Markov chains
- Applications and implementation of solution methods for Markov chains
- Method of supplementary variables
- Proxel simulation and phase distributions
- Modeling with hidden models
- Programming solution methods for different model classes
- Modeling and solving problems from medicine and engineering

Workload:

56 h lectures and exercise classes

124 h independent work

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
Solutions to Assignments 1-6 need to be handed in before the exam registration.	Oral Exam 30 Minutes	• Lecture 2 SWS • Exercise class 2 SWS

Prerequisites according to examination regulations:	Recommended prerequisites:
none	• Engineering mathematics • Computer programming

Media:	Literature:
recorded lectures and exercises are available	

Comments:

Ausgewählte Algorithmen der Computergraphik

(engl. Selected Algorithms in Computer Graphics)

Modulnummer: FIN-INF-120272

Link zum LSF:	LSF
Verantwortung:	Christian Rössl
Dozent:in:	Christian Rössl
Lehrveranstaltungen:	• Vorlesung <i>Ausgewählte Algorithmen der Computergraphik</i> • Übung <i>Ausgewählte Algorithmen der Computergraphik</i>
Verwendbarkeit:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. VC: Visual Computing

Kürzel
AACG

CP
6

Semester
Winter

Fachsem.
ab 1.

Dauer
1 Semester

Sprache
deutsch

Niveau
Master

Angestrebte Lernergebnisse:

Die Studierenden

- verstehen grundlegende Konzepte der Linearen Algebra und Numerik
- wenden diese an, um Algorithmen aus dem Bereich der Computergraphik zu implementieren
- erstellen eigene Implementierungen, Tests und Benchmarks
- können geeignete Werkzeuge (z.B. Programmiersprache, Bibliotheken) beurteilen, sinnvoll auswählen und anwenden

- können wissenschaftlichen Artikel im Bereich Computergraphik insbesondere im Hinblick auf mathematische Inhalte verstehen und implementieren

Inhalt:

- Linear least-squares approximation
- Data interpolation and approximation
- Matrix factorization
- Sparse matrices
- Regularization
- Implementation issues
- General applications and case studies
- The specific contents may vary each semester.

Die Veranstaltung kann je Bedarf in englischer oder deutscher Sprache abgehalten werden. Vorlesungsmaterialien stehen auf englisch zur Verfügung.

Arbeitsaufwand:

- 56 h Präsenz
- 124 h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
Um an der Prüfung teilnehmen zu können, sind Prüfungsvorleistungen erforderlich. Details werden in der ersten Vorlesung bekannt gegeben.	Mündliche Prüfung 20 Minuten	• Vorlesung 2 SWS • Übung 2 SWS

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
--	------------------------------------

keine	• Vorlesungen im Bereich Computergraphik z.B. ◦ Grundlagen der Computergraphik ◦ CAGD ◦ Mesh Processing • Spaß an mathematischen Aufgabenstellungen und Programmierung • Programmierkenntnisse in C++ oder Julia oder Matlab <i>und/oder</i> die Bereitschaft sich schnell und selbständig in neues einzuarbeiten • Grundlegende Kenntnisse der Versionsverwaltung mit Git • Bereitschaft, der Veranstaltung auch in englischer Sprache zu folgen
-------	--

Medienformen:	Literatur:
	Literatur wird in der Vorlesung bekannt gegeben.

Hinweise:

Wir verwenden i.d.R. Julia (<https://julialang.org>) als Programmiersprache (statt vormals Matlab und GNU Octave) und dazu C++ nach Bedarf. Wo genau der Fokus liegt, tarieren wir soweit möglich in der Veranstaltung je nach Vorkenntnissen und Wünschen aus.

Clean Code Development

(in German: Clean Code Development)

Module-ID: FIN-INF-120432

Link:	LSF
Responsibility:	Prof. Dr. Frank Ortmeier
Lecturer:	Prof. Dr. Frank Ortmeier
Classes:	• Vorlesung <i>Clean Code Development</i> • Übung <i>Clean Code Development</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INF: Schlüssel- und Methodenkompetenzen - M.Sc. INGINF: Informatik - M.Sc. INGINF: Schlüssel- und Methodenkompetenzen - M.Sc. DKE: Applied Data Science - M.Sc. DE: Methoden des Digital Engineering - M.Sc. DE: Methoden der Informatik - M.Sc. DE: Fachliche Spezialisierung - M.Sc. VC: Computer Science - M.Sc. VC: Schlüssel- und Methodenkompetenzen

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
CCD	6	Winter	ab 1.	1 Semester	english	Master

Intended learning outcomes:

Students:

- know and understand the principles of Clean Code Development
- are able to apply the associated tools and practices in practice
- learn to independently select and adapt the concepts suitable for a project
- learn to distribute tasks within a team and collaborate in an organized manner

Content:

- Software engineering deals with the creation and development of software, the organization and modeling of related data structures, and the operation of software systems. It thus encompasses software design, implementation, and management.
- The introductory lecture *Software Engineering I* provided the foundational knowledge for designing and writing good software.
- This advanced lecture introduces modern techniques and methods that are commonly used in the development of large software systems.
- We align with the four core values of *Clean Code Development* – evolvability, correctness, production efficiency, and comprehensibility.
- To achieve these goals, a variety of programming principles and practices will be introduced.
- These include, among others, team formation and organization in software development, principles and tools of Clean Code Development, continuous integration and automated build systems, bug tracking, error localization and debugging, automated and model-based testing, code analysis and quality metrics, requirements engineering and tracing, as well as distributed and component-based software architectures.
- The course will offer a high degree of practical engagement with tools and software development techniques, especially in the exercises. The goal is not only to convey abstract knowledge but also to develop hands-on competencies.
- Participation in the exercises is mandatory and will be conducted in groups.

Workload:

56h Präsenzzeit + 124h selbständige Arbeit am Projekt

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	Projekt	• Vorlesung (2 SWS) • Übung (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	Software Engineering

Media:	Literature:

Comments:

Computer-Assisted Surgery

(in German: Computer-Assisted Surgery)

Module-ID: FIN-INF-140003

Link:	LSF
Responsibility:	Christian Hansen
Lecturer:	Christian Hansen
Classes:	• Lecture <i>Computer-Assisted Surgery</i> • Seminar <i>Computer-Assisted Surgery</i> • Team project <i>Computer-Assisted Surgery</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. VC: Visual Computing

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
CAS	6	Winter	ab 1.	1 Semester	english	Master

Intended learning outcomes:

Students who complete this module ...

- have an in-depth knowledge of modern surgical and interventional imaging technologies such as iMRI, iCT and IOUS, as well as medical image processing
- understand methods such as surgical tracking, calibration and registration, which are essential for the development and application of surgical assistance systems
- can design and develop surgical planning and navigation systems.
- can write a state of the art report in the field of computer assisted surgery
- can read, analyse and briefly summarise scientific papers in a scientific report.

Within this module students can choose between a seminar or a team project. If students choose to undertake a seminar, they ...

- will be able to read, understand and present scientific publications in the field of computer-assisted surgery
- will understand how to critically analyse scientific publications and improve their ability to present the results of others' research.

If students opt for a team project, they ...

- will develop, test and evaluate a software tool in an interdisciplinary team
- will learn how to deal with conflict in interdisciplinary teams.

Content:

This lecture provides an overview of computer-assisted surgery. After an introduction to the basics, the state of the art in computer-assisted surgery is presented using clinical examples. The following topics will be covered:

- Fundamentals of intra-operative imaging
- Fundamentals of surgical visualisation
- Computer-assisted surgical planning
- Surgical navigation systems
- Surgical Augmented Reality
- Surgeon-Computer Interaction
- Robotic Surgery
- Medical Software Development and Evaluation

Workload:

56 h contact hours + 124 h self study

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	Oral exam	• Lecture (2SWS) • Seminar (2SWS) or team project (2SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	

Media:	Literature:

Comments:

Data Mining II - Advanced Topics in Data Mining

(in German: Data Mining II - Advanced Topics in Data Mining)

Module-ID: FIN-INF-120455

Link:	LSF
Responsibility:	Myra Spiliopoulou
Lecturer:	Myra Spiliopoulou
Classes:	Vorlesung DM 2 Übung DM 2 COURTESY TRANSLATION: • Lecture class DM2 • Exercise class DM2
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. WIF: Informatik - M.Sc. DKE: Learning Methods and Models for Data Science - M.Sc. DE: Fachliche Spezialisierung - M.Sc. VC: Computer Science

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
DM2	6	Winter	ab 1.	1 Semester	english	Master

Intended learning outcomes:

When successfully completing this module, the students:

- comprehend why temporal data need different learning algorithms and evaluation procedures than used on static data
- comprehend the behaviour of supervised, unsupervised and semi-supervised learning algorithms on temporal data

- can design and apply simple learning algorithms and workflows on temporal data and interpret the induced models
- can evaluate models - once and in continuous evaluation, since both are needed in temporal learning

and have thus acquired skills they need in order to design and evaluate temporal learning algorithms themselves

Content:

Block 1: Data Streams

- Basics
- Stream clustering: methods and evaluation approaches
- Stream classification: learning methods and concept drift detectors; evaluation approaches
- Semi-supervised stream learning: emthods and evaluation approaches

Block 2: Time series

- Basics
- Prediction methods
- Evaluation of predictors

Workload:

56 h Präsenz + 124 h selbstständige Arbeit **COURTESY TRANSLATION:** 56 h in class + 124 h self-study

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	• Klausur 120 Minuten • Prüfungszulassungsvoraussetzung: Erreichen einer minimalen Anzahl von Punkten durch Votierung auf Übungsaufgaben COURTESY TRANSLATION: • Written exam of the form 'Klausur' with a duration of 120 min • Prerequisite for the written exam: a minimum number of points must be achieved during the exercise classes; this procedure is called 'Votierung'	• Vorlesung (2 SWS) • Übung (2 SWS) COURTESY TRANSLATION: • Lecture (2 hours per week of the semester) • Exercise (2 hours per week of the semester)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	• Familiarity with learning algorithms for static tabular data • Familiarity with methods for the evaluation of models induced on static data

Media:	Literature:
---------------	--------------------

Block 1:

- Gama, J. (2010) *Knowledge discovery from data streams*. CRC Press.
- Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., Carvalho, A. C. D., & Gama, J. (2013). Data stream clustering: A survey. *ACM Computing Surveys (CSUR)*, 46(1), 1-31.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4), 1-37.
- Lemaire, V., Salperwyck, C., & Bondu, A. (2014, July). A survey on supervised classification on data streams. In *European Business Intelligence Summer School* (pp. 88-125). Springer, Cham.
- Žliobaitė, I., Bifet, A., Read, J., Pfahringer, B., & Holmes, G. (2015). Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, 98(3), 455-482.
- Murilo Gomes, H., Grzenda, M., Mello, R., Read, J., Nguyen, M. H. L., and Bifet, A. (2022). A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Computing Surveys*, 55(4). - **for Block 1D**

Block 2:

- Parmezan, A. R. S., Souza, V. M., & Batista, G. E. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences*, 484, 302-337.
- Cismondi, F. C., Fialho, A., Vieira, S., Reti, S., Sousa, J., Finkelstein, S. (2013). Missing data in medical databases: Impute, delete or classify? *Artificial Intelligence in Medicine*.
<http://dx.doi.org/10.1016/j.artmed.2013.01.003>.
- Further citations at the individual units: see the slidesets of the units.

Comments:

Data Science with Python

(in German: Data Science with Python)

Module-ID: FIN-INF-

Link:	LSF
Responsibility:	Dr. Christian Beyer
Lecturer:	Dr. Christian Beyer
Classes:	Data Science with Python
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INF: Schlüssel- und Methodenkompetenzen - M.Sc. INGINF: Informatik - M.Sc. INGINF: Schlüssel- und Methodenkompetenzen - M.Sc. WIF: Informatik - M.Sc. WIF: Schlüssel- und Methodenkompetenzen - M.Sc. DKE: Applied Data Science - M.Sc. VC: Computer Science - M.Sc. VC: Schlüssel- und Methodenkompetenzen - M.Sc. DE: Interdisziplinäres Team-Projekt

Abbreviation	Credit Points	Semester	Term	Duration	Language	Level
DSWP	6	Winter	2.	1 Semester	english	Master

Intended learning outcomes:

The course is about learning from data to perform predictions and obtain useful insights. In the seminar, we will use the programming language Python.

Necessary skills to manage and analyze data will be taught and practiced on real-world applications. Programming knowledge from other courses is helpful but not mandatory. However, students are expected to have a profound knowledge of fundamental data-analysis techniques, such as classification, regression and clustering. After successful completion of this course, the student will be able to perform the following tasks in Python:

- Import and preprocess raw data (files, databases, web APIs)
- Transform data for modelling

- Perform exploratory data analysis with summary statistics and visualization
- Understand, build and evaluate predictive classification and regression models, including tree-based models, ensembles and boosted models
- Communicate and disseminate results and findings through reproducible documents, presentations, websites and interactive web applications

Content:

Part Fundamentals & Visualization: Basics, scripts, workflows, vectors & functions in Python
 Explorative data visualization Data transformation Part Data Management & Exploratory Data Analysis: Data cleaning & scraping Generating hypotheses and an intuition about the data with exploratory data analysis Data import Data management Relational data Strings, categorical data, dates & time Iteration: imperative & functional programming Part Modeling: Linear regression Classification Evaluation Model selection & regularization (LASSO, Ridge) Feature selection & model interpretation Decision trees Ensembles: random forests Boosting: gradient boosted trees Unsupervised learning, e.g. k-means, hierarchical clustering, self-organizing maps, principal component analysis Topic modeling with simple graphical models Statistical testing Part Communication: Communication and dissemination of results through visualization and interpretable summaries with documents, notebooks, presentations & websites

Workload:

Attendance time = 28 h: - 2 SWS weekly seminar; Independent work outside the actual Seminar time = 152 h: - 76 h preparation and follow-up of the seminar topics - 76 h solving the tasks, incl. work in the laboratory 180h = 28h attendance time + 152h independent work

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	Project with presentation and project report	Seminar (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	Area 1: Data Mining, Machine Learning, Artificial Intelligence Area 2: Databases Area 3: Programming Languages and Software Engineering Area 4: Stochastics, Applied Statistics

Media:	Literature:
	Will provided during the seminar

Comments:

Module im Bereich: Ingenieurinformatik

AMS Lab Projects

(in German: AMS Lab Projects)

Module-ID: FIN-INF-999912

Link:	LSF
Responsibility:	Prof. Dr.-Ing. Benjamin Noack
Lecturer:	Prof. Dr.-Ing. Benjamin Noack
Classes:	<i>AMS Lab Projects</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INGINF: Informatik - M.Sc. INGINF: Ingenieurinformatik - M.Sc. WIF: Informatik - M.Sc. DKE: Applied Data Science - M.Sc. DE: Interdisziplinäres Team-Projekt - M.Sc. DE: Digital Engineering-Projekt - M.Sc. VC: Computer Science

Abbreviation AMS-Lab	Credit Points 6	Semester every	Term 2.	Duration 1 Semester	Language english	Level Master
--------------------------------	---------------------------	--------------------------	-------------------	-------------------------------	----------------------------	------------------------

Intended learning outcomes:

Students who complete the course ...

- can collaborate in teams to solve challenges in technical domains and deepen their technical expertise.
- can manage a project and define goals and milestones
- can acquire the knowledge in a technical domain to derive solutions
- can present and document their work and achievements

Content:

This module offers different team projects and individual projects on current, research-related

areas like:

- robotics
- sensor data processing
- navigation and planning
- optimization and control
- and more

Workload:

180 h of individual and team work

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	• Software and Documentation • Project report • Two Presentations	Project (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	

Media:	Literature:
	Literature will be announced in the lecture.

Comments:

Module im Bereich:
Schlüssel- und
Methodenkompetenzen

Clean Code Development

(in German: Clean Code Development)

Module-ID: FIN-INF-120432

Link:	LSF
Responsibility:	Prof. Dr. Frank Ortmeier
Lecturer:	Prof. Dr. Frank Ortmeier
Classes:	• Vorlesung <i>Clean Code Development</i> • Übung <i>Clean Code Development</i>
Applicability in curriculum:	- M.Sc. INF: Informatik - M.Sc. INF: Schlüssel- und Methodenkompetenzen - M.Sc. INGINF: Informatik - M.Sc. INGINF: Schlüssel- und Methodenkompetenzen - M.Sc. DKE: Applied Data Science - M.Sc. DE: Methoden des Digital Engineering - M.Sc. DE: Methoden der Informatik - M.Sc. DE: Fachliche Spezialisierung - M.Sc. VC: Computer Science - M.Sc. VC: Schlüssel- und Methodenkompetenzen

Abbreviation
CCD

Credit Points
6

Semester
Winter

Term
ab 1.

Duration
1 Semester

Language
english

Level
Master

Intended learning outcomes:

Students:

- know and understand the principles of Clean Code Development
- are able to apply the associated tools and practices in practice
- learn to independently select and adapt the concepts suitable for a project
- learn to distribute tasks within a team and collaborate in an organized manner

Content:

- Software engineering deals with the creation and development of software, the organization and modeling of related data structures, and the operation of software systems. It thus encompasses software design, implementation, and management.
- The introductory lecture *Software Engineering I* provided the foundational knowledge for designing and writing good software.
- This advanced lecture introduces modern techniques and methods that are commonly used in the development of large software systems.
- We align with the four core values of *Clean Code Development* – evolvability, correctness, production efficiency, and comprehensibility.
- To achieve these goals, a variety of programming principles and practices will be introduced.
- These include, among others, team formation and organization in software development, principles and tools of Clean Code Development, continuous integration and automated build systems, bug tracking, error localization and debugging, automated and model-based testing, code analysis and quality metrics, requirements engineering and tracing, as well as distributed and component-based software architectures.
- The course will offer a high degree of practical engagement with tools and software development techniques, especially in the exercises. The goal is not only to convey abstract knowledge but also to develop hands-on competencies.
- Participation in the exercises is mandatory and will be conducted in groups.

Workload:

56h Präsenzzeit + 124h selbständige Arbeit am Projekt

Pre-examination requirements:	Type of examination:	Teaching method / lecture hours per week (SWS):
	Projekt	• Vorlesung (2 SWS) • Übung (2 SWS)

Prerequisites according to examination regulations:	Recommended prerequisites:
keine	Software Engineering

Media:	Literature:

Comments:

Masterarbeit

Feedback

Die Modulhandbücher der FIN wurden auf ein neues System umgestellt. Wir freuen uns über Feedback dazu.

Ja, ich habe Vorschläge.

□□ Credits

Die maßgeblichen Beteiligten waren:

Lukas Eichel (□□♂ Code-Held)

Christos Lachanas (□□♂ Code-Held)

Michael Preuß (□□♂ Technik-Held 1, um aus einer Testinstanz eine Lösung zu machen)

Heiko Dorwarth (□□♂ Technik-Held 2 für ein flüssiges Gitlab und Bookstack)

Graham Horton (□□♂ Aktualisierungs-Held)

Ich danke euch für dieses Engagement! □□

Jana (03.09.2025)